sensors

Special Issue Reprint

Image and Video Processing and Recognition Based on Artificial Intelligence

3rd Edition

Edited by
Kang Ryoung Park, Sangyoun Lee and Euntai Kim

mdpi.com/journal/sensors

MDPI

Image and Video Processing and Recognition Based on Artificial Intelligence: 3rd Edition

Image and Video Processing and Recognition Based on Artificial Intelligence: 3rd Edition

Guest Editors

Kang Ryoung Park
Sangyoun Lee
Euntai Kim

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Kang Ryoung Park
Division of Electronics and Electrical Engineering
Dongguk University
Seoul
Republic of Korea

Sangyoun Lee
School of Electrical and Electronic Engineering
Yonsei University
Seoul
Republic of Korea

Euntai Kim
School of Electrical and Electronic Engineering
Yonsei University
Seoul
Republic of Korea

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Sensors* (ISSN 1424-8220), freely accessible at: https://www.mdpi.com/journal/sensors/special_issues/8P92P589Z0.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-6726-4 (Hbk)
ISBN 978-3-7258-6727-1 (PDF)
https://doi.org/10.3390/books978-3-7258-6727-1

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

About the Editors . . . . . **vii**

Jae-Hyuk Park, Mohamed Mahmoud and Hyun-Soo Kang
Conv3D-Based Video Violence Detection Network Using Optical Flow and RGB Data
Reprinted from: *Sensors* **2024**, *24*, 317, https://doi.org/10.3390/s24020317 . . . . . **1**

HeanSung Lee, Minjung Kim, Sungjun Jang, Han Byeol Bae and Sangyoun Lee
Multi-Granularity Aggregation with Spatiotemporal Consistency for Video-Based Person Re-Identification
Reprinted from: *Sensors* **2024**, *24*, 2229, https://doi.org/10.3390/s24072229 . . . . . **16**

Feng Huang, Yunxiang Li, Xiaojing Ye and Jing Wu
Infrared Image Super-Resolution Network Utilizing the Enhanced Transformer and U-Net
Reprinted from: *Sensors* **2024**, *24*, 4686, https://doi.org/10.3390/s24144686 . . . . . **37**

Tomoyuki Suzuki and Yoshimitsu Aoki
RetinaViT: Efficient Visual Backbone for Online Video Streams
Reprinted from: *Sensors* **2024**, *24*, 5457, https://doi.org/10.3390/s24175457 . . . . . **63**

Gang Wan, Zhijuan Su, Yitian Wu, Ningbo Guo, Dianwei Cong, Zhanji Wei, et al.
High-Precision Multi-Object Tracking in Satellite Videos via Pixel-Wise Adaptive Feature Enhancement
Reprinted from: *Sensors* **2024**, *24*, 6489, https://doi.org/10.3390/s24196489 . . . . . **84**

Jiho Choi and Sang Jun Lee
Neural Radiance Fields for Fisheye Driving Scenes Using Edge-Aware Integrated Depth Supervision
Reprinted from: *Sensors* **2024**, *24*, 6790, https://doi.org/10.3390/s24216790 . . . . . **103**

Mst Rumana Sumi, Priyanka Das, Afzal Hossain, Soumyabrata Dey and Stephanie Schuckers
A Comprehensive Evaluation of Iris Segmentation on Benchmarking Datasets
Reprinted from: *Sensors* **2024**, *24*, 7079, https://doi.org/10.3390/s24217079 . . . . . **115**

Seunghyun Baik and Euntai Kim
Detection of Human Traffic Controllers Wearing Construction Workwear via Synthetic Data Generation
Reprinted from: *Sensors* **2025**, *25*, 816, https://doi.org/10.3390/s25030816 . . . . . **134**

Josue Lopez-Cabrejos, Thuanne Paixão, Ana Beatriz Alvarez and Diodomiro Baldomero Luque
An Efficient and Low-Complexity Transformer-Based Deep Learning Framework for High-Dynamic-Range Image Reconstruction
Reprinted from: *Sensors* **2025**, *25*, 1497, https://doi.org/10.3390/s25051497 . . . . . **156**

Jie Li, Lei Zhao, Heng Li, Xiaojun Xue and Hui Liu
MixRformer: Dual-Branch Network for Underwater Image Enhancement in Wavelet Domain
Reprinted from: *Sensors* **2025**, *25*, 3302, https://doi.org/10.3390/s25113302 . . . . . **176**

Yuki Kawawaki and Yuji Yamakawa
High-Speed Multiple Object Tracking Based on Fusion of Intelligent and Real-Time Image Processing
Reprinted from: *Sensors* **2025**, *25*, 3400, https://doi.org/10.3390/s25113400 . . . . . **199**

Jun Yin, Fei Wu, Hao Su, Peng Huang and Yuetong Qixuan
Improvement of SAM2 Algorithm Based on Kalman Filtering for Long-Term Video Object Segmentation
Reprinted from: *Sensors* **2025**, *25*, 4199, https://doi.org/10.3390/s25134199 . . . . . **225**

Chajin Shin, Yonghwan Kim, KwangPyo Choi and Sangyoun Lee
Masked Feature Residual Coding for Neural Video Compression
Reprinted from: *Sensors* **2025**, *25*, 4460, https://doi.org/10.3390/s25144460 . . . . . **246**

Jakub Kwaśniak, Mateusz Majtka, Mateusz Lorkiewicz, Tomasz Grajek and Krzysztof Klimaszewski
Advanced ANN Architecture for the CTU Partitioning in All Intra HEVC
Reprinted from: *Sensors* **2025**, *25*, 5971, https://doi.org/10.3390/s25195971 . . . . . **268**

Marta Bistroń, Jacek M. Żurada and Zbigniew Piotrowski
Deep Learning for Image Watermarking: A Comprehensive Review and Analysis of Techniques, Challenges, and Applications
Reprinted from: *Sensors* **2026**, *26*, 444, https://doi.org/10.3390/s26020444 . . . . . **284**

About the Editors

Kang Ryoung Park

Kang Ryoung Park received his B.S. and M.S. degrees in Electronic Engineering from Yonsei University, Seoul, South Korea, in 1994 and 1996, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from Yonsei University in 2000. He has been a professor in the Division of Electronics and Electrical Engineering at Dongguk University since March 2013. His research interests include image processing and deep learning.

Sangyoun Lee

Sangyoun Lee received his B.S. and M.S. degrees in Electronic Engineering from Yonsei University, Seoul, South Korea, in 1987 and 1989, respectively, and Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1999. He is currently a Professor at the School of Electrical and Electronic Engineering and the Head of the Image and Video Pattern Recognition Laboratory, Yonsei University. His research interests include all aspects of computer vision, with a special focus on pattern recognition for face detection and recognition, advanced driver-assistance systems, and video codecs.

Euntai Kim

Euntai Kim received his B.S., M.S., and Ph.D. degrees in Electronic Engineering, all from Yonsei University, Seoul, Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2002, he was a Full-Time Lecturer in the Department of Control and Instrumentation Engineering, Hankyong National University, Kyonggi-do, Korea. Since 2002, he has been with the faculty of the School of Electrical and Electronic Engineering, Yonsei University, where he is currently a professor. In 2008, he was a Visiting Researcher with the Berkeley Initiative in Soft Computing, University of California, Berkeley, CA, USA. He was also a Visiting Researcher with the Korea Institute of Science and Technology (KIST), Korea, in 2018. His current research interests include computational intelligence and statistical machine learning and deep learning and their application to intelligent robotics, autonomous vehicles, and robot vision.

sensors

MDPI

Article

Conv3D-Based Video Violence Detection Network Using Optical Flow and RGB Data

Jae-Hyuk Park [1], Mohamed Mahmoud [1,2] and Hyun-Soo Kang [1,*]

[1] Department of Information and Communication Engineering, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju-si 28644, Republic of Korea; 2022234002@cbnu.ac.kr (J.-H.P.); mohamedabokhalil@aun.edu.eg (M.M.)
[2] Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut 71515, Egypt
* Correspondence: hskang@cbnu.ac.kr

Abstract: Detecting violent behavior in videos to ensure public safety and security poses a significant challenge. Precisely identifying and categorizing instances of violence in real-life closed-circuit television, which vary across specifications and locations, requires comprehensive understanding and processing of the sequential information embedded in these videos. This study aims to introduce a model that adeptly grasps the spatiotemporal context of videos within diverse settings and specifications of violent scenarios. We propose a method to accurately capture spatiotemporal features linked to violent behaviors using optical flow and RGB data. The approach leverages a Conv3D-based ResNet-3D model as the foundational network, capable of handling high-dimensional video data. The efficiency and accuracy of violence detection are enhanced by integrating an attention mechanism, which assigns greater weight to the most crucial frames within the RGB and optical-flow sequences during instances of violence. Our model was evaluated on the UBI-Fight, Hockey, Crowd, and Movie-Fights datasets; the proposed method outperformed existing state-of-the-art techniques, achieving area under the curve scores of 95.4, 98.1, 94.5, and 100.0 on the respective datasets. Moreover, this research not only has the potential to be applied in real-time surveillance systems but also promises to contribute to a broader spectrum of research in video analysis and understanding.

Keywords: deep learning; CCTV anomaly detection; optical flow; attention network

1. Introduction

Identifying violent conduct within video content has quickly become critical in public safety and security research. In the society of today, the widespread integration of closed-circuit television (CCTV) for crime prevention and surveillance has reached unprecedented levels. However, this expansion in CCTV deployment has not been matched by a corresponding increase in human resources for the monitoring and oversight of these systems. Consequently, automated systems that can rapidly and precisely detect instances of violence in real time are vital. These systems can provide a timely alert to hazardous scenarios, facilitating swift responses.

Traditional methods for detecting violence predominantly concentrate on manual feature extraction from video data [1–3]. However, these approaches frequently exhibit shortcomings in robustness and adaptability when deployed in real-world settings. This limitation stems from the intricate interplay of factors such as installation angles, diverse locations, varying backgrounds, and video resolutions of CCTV footage. In recent times, propelled by advances in artificial intelligence, several deep-learning models have been developed. These models autonomously identify features and patterns that were previously challenging to detect using conventional violence-detection methods [4–7].

Assessing the extent of motion plays a pivotal role in identifying violent occurrences in video content. Optical-flow techniques are commonly used to capture such motion

patterns. This approach quantifies the movement between successive frames, proving effective in swiftly evolving scenes or instances of violence. However, optical-flow frames alone may not comprehensively detect violent scenarios. RGB frames offer indispensable cues to recognize violence within static images, revealing details about the elements in the scene and their interactions. For instance, a motion like running might be construed as benign, as what may be jogging in one context could signify a potentially hazardous situation in another. Herein lies the significance of RGB data. Analyzing color and object patterns within the frames enables a violence-detection model to comprehend a wider context, thereby minimizing false alarms. The insights obtained from the fusion of optical flow and RGB data are mutually complementary. Concurrently integrating both types of data can significantly bolster the efficiency and precision of violence detection by providing a more comprehensive understanding of the content.

For successful integration of optical flow and RGB data, a model that adeptly comprehends the interconnections between temporal and spatial dimensions becomes imperative. Convolutional neural networks (CNNs) have demonstrated their efficacy in handling spatial data in various applications, such as image recognition [8–11], alongside applications in security and image inpainting [12,13]. In the realm of video analysis, 3D CNN (Conv3D) models have gained prominence. These models consider the width, height, and temporal sequences (frames) of images, effectively handling both the temporal and spatial information inherent in videos. This concurrent processing of spatiotemporal aspects marks the capability of Conv3D models in video analysis and recognition.

This study introduces a robust violence-detection network tailored for diverse backgrounds. We focus on isolating and scrutinizing moving objects within videos rather than the static background, thereby enhancing detection capabilities across various datasets that portray violent scenarios. To capture features associated with non-background moving objects, we harness optical-flow features extracted from the video and complement these with RGB frames to comprehensively interpret the visual context. Both components serve as inputs to the model. Our methodology integrates these combined features to understand the spatiotemporal characteristics embedded within the video frames. To achieve this, we employ a Conv3D-based ResNet model, known as ResNet-3D, renowned for its proficiency in comprehending video data. This specialized model allows our system to be trained to recognize patterns of violence, considering both dynamic movements and visual context. By capitalizing on the strengths of optical flow and RGB data features, our approach represents a significant step towards developing a more accurate and reliable violence-detection model. This model surpasses existing models by addressing the challenges posed by the varying specifications of real-world CCTV setups, thereby offering enhanced robustness and adaptability.

The key contributions outlined in our study are as follows:

1. Extraction of motion information between frames using optical-flow features;
2. Understanding of visual contexts, such as objects and interactions, using RGB data;
3. Development of a model that captures the relationship between integrated RGB and optical-flow frames, and understands both spatial and temporal dimensions, using ResNet-3D and an attention module.

The remainder of the paper is structured as follows. Section 2 provides an in-depth exploration of crucial technologies fundamental to developing a violence-detection system. Section 3 outlines the algorithm designed to proficiently detect instances of violence. Section 4 emphasizes the presented data and preparatory steps for training, covering aspects such as loss functions and evaluation metrics. Section 5 evaluates the performance of the model using CCTV data. Finally, Section 6 concludes the paper, summarizing the research findings of this paper.

2. Related Work

In this section, we provide an overview of violence-detection techniques and focus on the preliminary studies for the method proposed in this paper. We discuss research on

optical flow, studies combining optical flow with RGB data, Conv3D, and the trends in video-violence detection research.

2.1. Utilization of Optical Flow in Violence Detection

Recently, the importance of using optical flow, which extracts critical information about the dynamic movements of objects in violent situations, has been increasingly highlighted. Early research in violence detection applied to human action recognition [14] used traditional computer-vision techniques to extract optical flow by employing the dense trajectories method, which calculates trajectories of uniformly sampled points across frames to capture extensive optical flow information. Such traditional methods had limitations in precisely extracting optical flows from videos with various resolutions or containing numerous objects, resulting in violence-detection models with low performance in terms of accuracy and speed.

To overcome the limitations of these traditional computer-vision techniques, various deep-learning models that efficiently extract optical flows from image frames have been developed. There are three types of studies: studies that use spatial pyramid networks to estimate optical flows [15], studies that assume that most scenes in videos consist of rigid objects and estimate optical flow in such environments [16], and those that apply the transformer architecture for optical-flow estimation [17]. These deep-learning models have demonstrated high action-recognition rates, which have been widely acknowledged. The trend in violence-detection research is continuing towards enhancing the efficiency and accuracy of video analysis using optical flow.

2.2. Combining Optical Flow with RGB Data

Recent research has focused on combining optical flow with RGB data to gain an integrated understanding of the motion characteristics and visual context within video data [18]. Integrating features from RGB and optical flow allows for capturing various aspects and characteristics of video data, enabling the model to better understand the complex structures or patterns within the data. Moreover, it prevents the model from relying too heavily on specific information and allows it to learn more general patterns based on diverse information, thus reducing the risk of overfitting.

Several deep-learning models use both optical flow and RGB data. For instance, the I3D model [18], designed for video classification and action recognition, has shown high performance in video recognition using optical flow and RGB simultaneously as inputs. In the video object recognition approach proposed by Zhang [19], issues such as motion blur, defocus, partial occlusion, and rare poses in videos were addressed by combining optical flow with RGB data. Studies that combined RNN [20] and CNN models to learn temporal features [19] also used both optical flow and RGB data. Additionally, various video-classification models have used RGB and optical flow [21–23].

2.3. Conv3D

Conv3D is gaining significant attention in the field of video processing, particularly in action recognition within videos. Unlike traditional CNNs, which focus on 2D data such as images, Conv3D is designed to detect both the static content of images and the dynamic changes over time. The key to this capability is that Conv3D performs convolutions over time and across width and height, which is typical in CNNs. Owing to this feature, Conv3D can simultaneously capture and learn the temporal continuity and spatial characteristics of videos.

Consequently, the amount of research that uses Conv3D for video data analysis is increasing. The TSN model [24], a new approach that more precisely represents motion within videos, is an example. TSN segments full videos into multiple parts and integrates the sampled frame information from each segment to derive a comprehensive representation of the videos. This approach has improved the accuracy of human-action recognition in videos. Christoph [25] introduced a ResNet-based Conv3D architecture, similar to the one

used in this study, to explore methods of processing spatiotemporal information in videos. Early Conv3D [26] application studies presented research that improved the way of learning features through convolutions in the spatiotemporal dimensions of videos. Another study [27] proposed the C3D model, exploring a methodology for directly learning spatiotemporal features from entire video clips. This model deeply analyzed the relationships and patterns between frames using continuous convolutions in the temporal dimension.

Thus, research centered around Conv3D is achieving high performance in action recognition while overcoming the spatiotemporal complexity of videos. This significantly expands its potential applications in various fields related to video processing.

2.4. Video-Violence Detection

Automatic detection of violent behaviors in videos has become important in various sectors, such as security, education, and transportation. In the early stages of video-violence detection research, features extracted manually from the videos were used to develop detection models. One of the seminal works proposed by Hansner [4] used handcrafted features to detect violent actions in crowds in real time. It is a representative example of early violence-detection models. Although these techniques achieved considerable recognition rates with simple features in specific video environments, they fell short in detecting violent patterns in the complex settings of various CCTV systems.

With advances in technologies such as deep learning, end-to-end models that extract features directly from video data are being proposed. These methods, with the significant increase in the amount of data available for model training, have shown much higher performance than traditional handcrafted-feature-based approaches and are highly accurate and reliable in real-world environments.

Various deep-learning-based violence-detection models, such as the Conv-AE [6] model, learn temporal consistency in video sequences by combining the feature extraction capability of CNNs with the data reconstruction ability of AutoEncoders [5] after compressing the data. Another example is Explainable VAD [7], a violence-detection network based on unsupervised learning that learns general knowledge from video data and detects abnormal events in specific contexts. Moreover, the skeleton-based approach [28] abstracts actions using human skeletons to form continuous trajectories in each frame and uses this information to detect anomalies in movement.

Research on video-violence detection has evolved from early handcrafted-feature-based methods to now leveraging deep learning to utilize various features comprehensively. This technological progression plays a crucial role in accurately detecting acts of violence in more diverse and complex environments.

3. Approach

We propose a violence-detection model that combines RGB and optical-flow data using a Conv3D model based on ResNet. The structure of the proposed violence-detection model is shown in Figure 1. Given video data, RGB frames are extracted at regular frame intervals. Subsequently, optical-flow frames are derived from these RGB frames using a deep-learning model; then, the RGB and optical-flow frames are combined. The combined frames are fed into the backbone model, Conv3D + attention module, for violence-detection classification. The specific details of the proposed model are described below.

3.1. Optical Flow

Optical flow represents the pixel movement between two consecutive image frames as a vector field and is crucial for detecting the movement of objects or the background, as well as camera motion. Violent actions are characterized by abrupt and unpredictable movements, which can be analyzed using optical flow. Understanding various patterns of movement during violent interactions through optical flow enables more accurate detection of the intensity and nature of the behavior. The precision and efficiency of the model in

violence detection can be enhanced using the pattern changes in optical flow corresponding to violent behaviors in different environments.

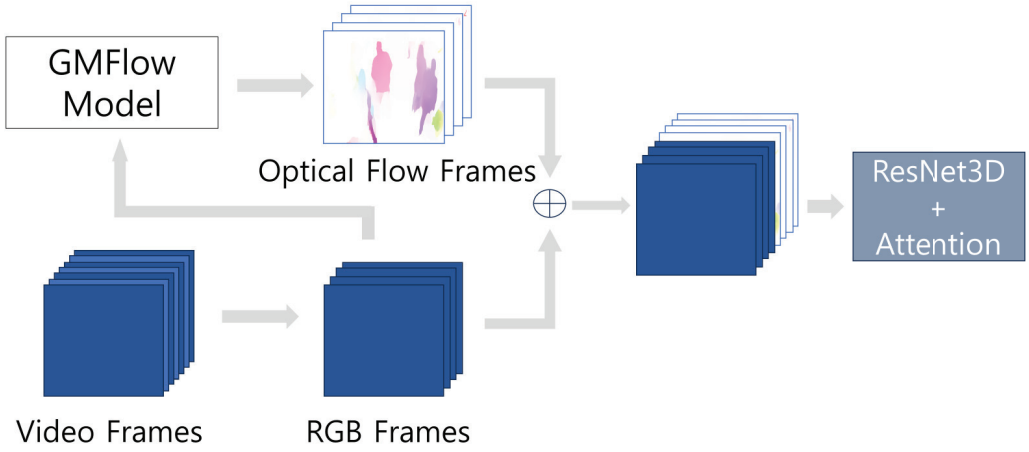


Figure 1. Proposed violence detection network.

We employed GMFlow [29], a state-of-the-art model that achieved the highest accuracy on the datasets used for validating optical-flow model performance, such as Sintel [30] and KITTI [31]. GMFlow uses a pre-trained model with the Sintel dataset, which consists of videos with various backgrounds and their corresponding optical flow. The model structure of GMFlow is shown in Figure 2. Figure 3 illustrates the resulting image of optical flow obtained from a video of a violent situation, emphasizing the dynamic changes between video frames. These highlight regions of movement, mainly detecting features of moving objects. Optical flow visually represents these patterns and directions of movement, aiding in a clearer understanding of changes in moving objects or scenes.

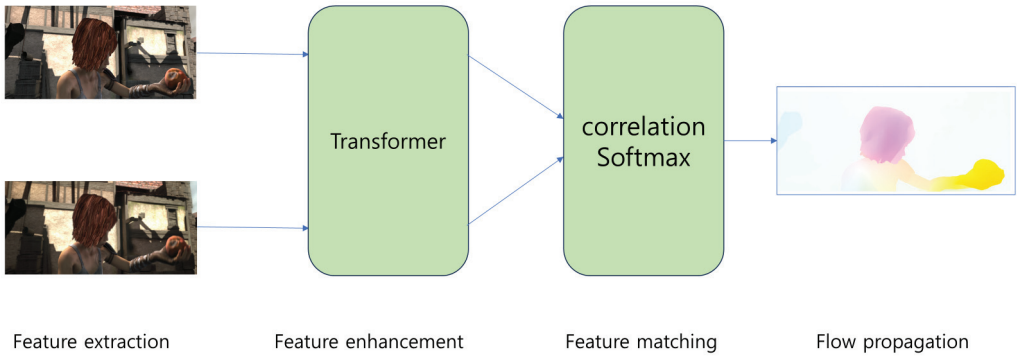


Figure 2. GMFlow model used for optical flow extraction.

3.2. RGB Data

In addition to optical flow, which captures dynamic movements in videos, RGB data should also be used to capture static context information effectively. RGB data contain the original color information of the video, which can identify static information such as the background, objects, and environment of violent behavior. Since violent situations are often closely linked to specific backgrounds or contexts, this static information provides critical insight for increasing the detection accuracy. For example, the act of one person pushing another could have a significantly different contextual meaning when occurring in a park, suggesting play, as opposed to in a parking lot, suggesting violence. Thus, RGB data provide the context of the action, helping to ascertain the intent or cause of action more accurately.

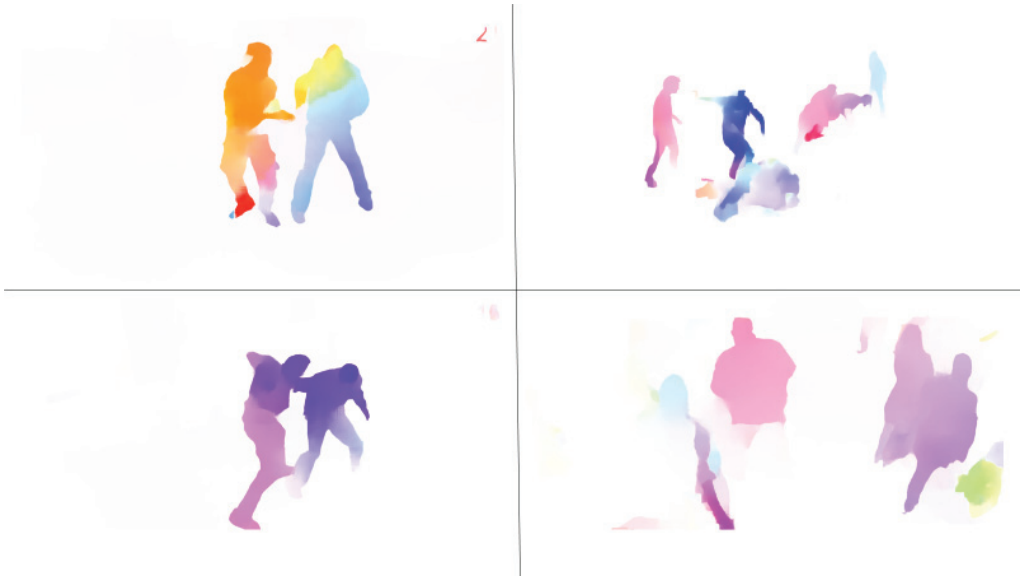


Figure 3. Optical-flow image extracted from a violent situation video.

3.3. Conv3D

Conv3D is a convolutional operation designed for 3D data, especially temporal data such as videos. While traditional CNNs only perform convolutions across the spatial dimensions of height and width of an image, Conv3D additionally considers the temporal dimension, performing convolutions across three dimensions: height, width, and time.

Figure 4 visualizes the operation of Conv3D. The process of Conv3D operation proceeds as follows:

1. The input is a 4D tensor of the form $C \times T \times W \times H$, where C represents the number of channels (3), T represents the temporal dimension (frames), H represents the height of the image, and W represents the width of the image.
2. For the convolution filter, a 5D tensor of the form $F \times C \times T \times H$ is used. Here, F is the number of filters. The depth of the filter must match the number of input channels.
3. When performing the convolution, a 3D filter is applied to each channel of the input data to generate output values. The filter slides along the temporal dimension, computing spatial convolutions at each position. These results are summed to obtain the final output value.
4. The output is a 4D tensor of the form $F \times T \times W \times H$, composed of the convolution results for each filter.

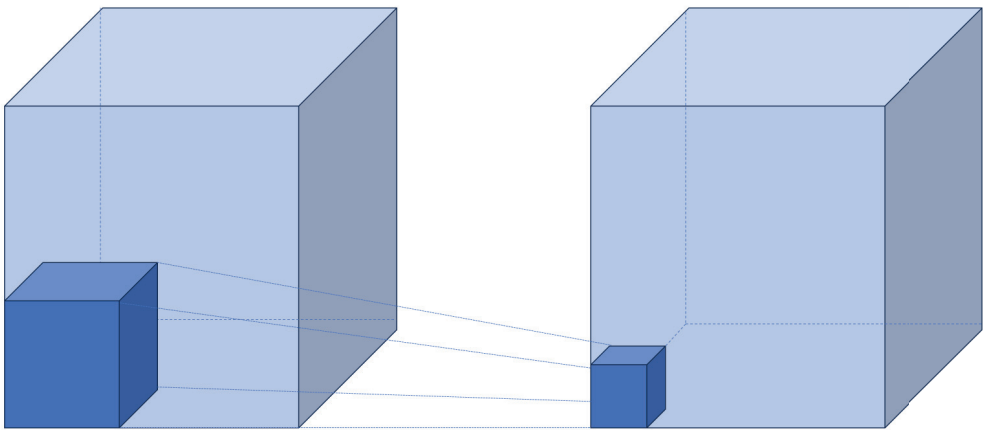


Figure 4. Conv3D calculation.

Through these 3D operations, Conv3D can extract not only the features of images but also the information between frames along the time axis.

3.4. ResNet-3D

ResNet-3D [25] is an architecture that extends the core principles of the existing CNN model, ResNet, and applies it to the Conv3D model. This allows the model to capture temporal continuity between video frames. Such a feature enables the model to effectively learn patterns and information in the temporal dimension.

The essential feature of ResNet, the residual connection, is also applied to this ResNet-3D structure. One of the main advantages of residual connections is that they alleviate the vanishing gradient problem, which is common in deep networks. This issue can become even more pronounced in complex networks that handle high-dimensional data, such as videos in the case of Conv3D. With the introduction of residual connections, stable improvements in model performance are possible even in deep Conv3D architectures.

Because ResNet-3D processes information in the temporal dimension, it can intricately learn information about dynamic characteristics such as movement, continuity, and temporal changes, demonstrating superior performance on video analysis problems like action recognition compared with other Conv3D models. Figure 5 below depicts the structure of the ResNet-3D model used in this study.

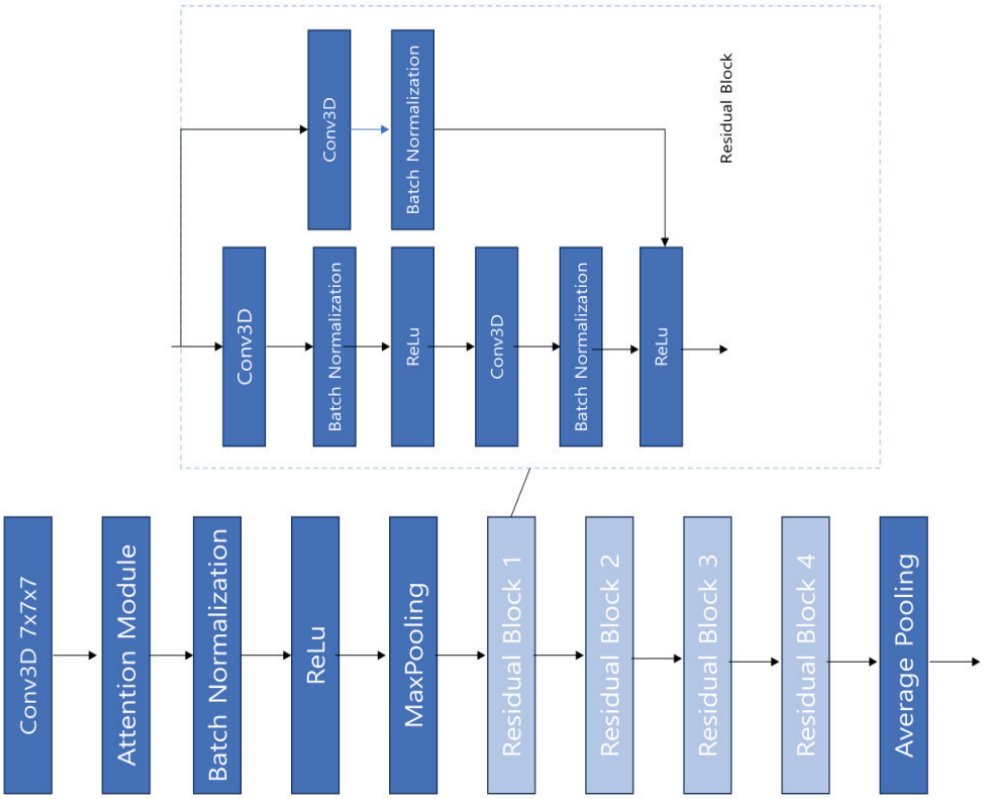


Figure 5. ResNet-3D model structure.

As shown in Figure 5, ResNet-3D uses residual connections to mitigate the vanishing gradient problem by adding skip connections that directly add the input to the output of each layer, aiding the network in learning more effectively. It also employs bottleneck structures at intervals to reduce the computational load and uses batch normalization to enhance network stability. We used ResNet-3D based on Conv3D as the backbone model, focusing on learning the temporal continuity of video data to extract more sophisticated features.

3.5. Attention Module

To detect violent situations, it is crucial to capture the dynamic changes between consecutive frames within the video and the significant frames related to dangerous visual information. We used data in the form of a combination of RGB frames and optical flow and employed the SEModule3D attention module within our backbone model, ResNet-3D, to assist the model in identifying important frames from the combined data. SEModule3D is an adaptation of the widely used squeeze-and-excitation (SE) attention module in images for 3D data, modeled to suit 3D information. The SE network was first proposed in [32], and its basic concept is to explicitly model the interdependencies between channels of feature maps. This allows the network to learn the importance of each Frame, thereby enhancing the representativeness of the network. The operation of SEModule3D is shown in Figure 6.

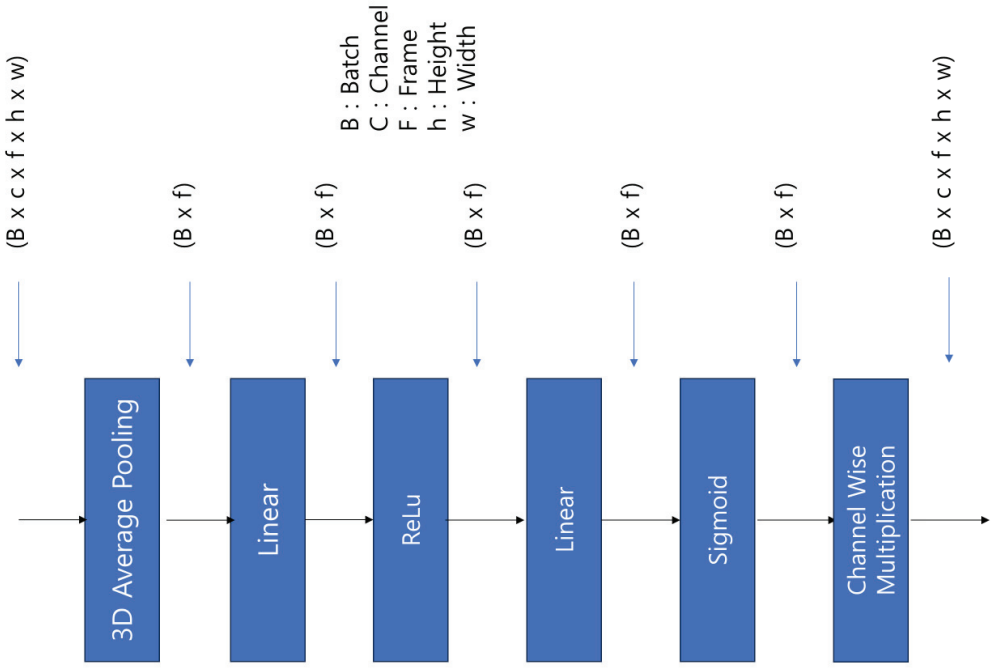


Figure 6. Attention module structure.

The operational structure of SEModule3D is as follows:

1. Squeeze: It uses average pooling to extract the global average information of each Frame.
2. Excitation: Based on the global average statistics, the feature map is transformed through two fully connected layers. The first layer reduces the number of Frames, followed by an adjustment to values between 0 and 1 via the rectified linear unit and the sigmoid functions.
3. Rescale: The final output is obtained by multiplying the input feature map with the sigmoid output. This process enables the network to learn the importance of each Frame, emphasizing important Frames and suppressing less important ones.

The attention module helps the violence-detection model to focus on important Frames within the input data, effectively capturing critical areas within specific frames and important movements between consecutive frames.

4. Experiment

In this section, we introduce the experimental datasets, describe the loss functions and optimization techniques employed, and provide an overview of the actual experimental setup.

4.1. Training Data

To train the model to detect violent behaviors, we used training data that include a variety of backgrounds and situations captured by real CCTV footage. Each dataset was divided into an 8:2 ratio for training and validation purposes. The datasets used are as follows:

UBI-FIGHT [33]: This contained real violent situations occurring in various places and consisted of 1000 data samples. Among these, 786 were normal situations and 214 were abnormal, making it an imbalanced dataset.

Hockey [34]: This contained violent situations from real hockey games, comprising 1000 data samples with 500 normal and 500 abnormal situations.

Movie-Fights [35]: This contained 200 violent situations captured in various backgrounds from movies—100 normal and 100 abnormal.

Crowd [4]: A video dataset about collective violence situations. It contained 256 data samples, with 123 normal and 123 abnormal situations.

4.2. Loss Function

For training, a loss function is used as a metric to gauge the prediction and error. A loss function refers to a function that calculates the error based on the difference between input and output values. Owing to the nature of the data used in anomaly detection, which often involves an imbalance between normal and abnormal data, the focal loss function [36] is employed to address this imbalance as much as possible. It aids the handling of imbalanced class distributions in binary and multi-class classification problems and performs effectively for datasets with imbalanced distributions, which are not adequately addressed by standard cross-entropy loss.

Focal loss is based on cross-entropy loss; however, it assigns greater weight to misclassified classes and reduces the loss for correctly classified classes. This encourages the model to focus on classes that are difficult to classify and less learned. Below is the focal loss formula.

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

The essence of focal loss is as follows. During the model training, if a sample is deemed easy, the value of p_t in the formula becomes high, and by adding $(1-p_t)^{\gamma}$ to the loss, a penalty is imposed for high-confidence predictions. Here, γ is known as the focusing parameter, and it serves to reduce the contribution of easy samples to the loss during model training.

4.3. Optimization Technique

We employed AdamW [37], an improved version of adaptive moment estimation (Adam) [38], to address the issue of overfitting caused by training data from imbalanced datasets.

4.3.1. Adam

Adam is one of the deep-learning optimization algorithms, which combines the advantages of momentum [39] and RMSprop [40]. Adam adjusts the learning rate for each parameter using the average of the past squared gradients.

Adam is an optimization method that adaptively adjusts the learning rate for each parameter, following the update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t, \tag{2}$$

- θ represents the parameters to be optimized;
- t is the current time step;
- η is the learning rate;
- $\hat{v}_t$ and $\hat{m}_t$ are bias-corrected estimates of the first and second moments of the gradients, respectively;

- ϵ is a small scalar added to improve numerical stability.

The bias-corrected estimates, $\hat{m}_t$ and $\hat{v}_t$, are computed as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{3}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{4}$$

where m_t and v_t are the first moments (the mean) and the second moment (the uncentered variance) of the gradients, and β_1 and β_2 are the decay rates for these moments.

4.3.2. AdamW

AdamW is a modification of the Adam update rule, incorporating L2 weight decay [41]. L2 weight decay is a regularization method that limits the complexity of the model to prevent overfitting. The update rule of AdamW is similar to that of Adam, but includes an additional weight decay term during the parameter update phase. The formula for AdamW is as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}(\hat{m}_t + \lambda\theta_t) \tag{5}$$

- θ denotes the parameters of the model;
- t indexes the current time step;
- η is the learning rate;
- $\hat{m}_t$ is the bias-corrected estimate of the first moment (the mean) of the gradients;
- $\hat{v}_t$ is the bias-corrected estimate of the second moment (the uncentered variance) of the gradients;
- ϵ is a small scalar used to prevent division by zero and ensure numerical stability;
- λ represents the weight decay coefficient.

Here, $\lambda\theta_t$ is the parameter for L2 weight decay, which prevents overfitting by subtracting a small value from the weight updates. Datasets like those for violence detection often consist of imbalanced data, where the number of normal instances far exceeds that of abnormal ones. Therefore, to mitigate the problem of overfitting due to training on imbalanced datasets, we employed the AdamW optimization technique.

4.3.3. Training Environment

The training was conducted on Windows 11 using an i7-13700K CPU, an NVIDIA GeForce RTX 4090 graphics card, and Pytorch 2.00 + CUDA 11.7 version.

4.3.4. Evaluation Metrics

Accuracy and the receiver operating characteristic–area under the curve (ROC-AUC) [42] were used as metrics to evaluate the performance of the model. Accuracy is one of the most intuitively used performance evaluation metrics in classification problems. It is calculated as the ratio of correct predictions to the total number of predictions. Below is the formula for accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

Here, True positive (TP) is when the actual class is positive and the model also predicts positive. True negative (TN) is when the actual class is negative and the model predicts negative. False positive (FP) is when the actual class is negative but the model predicts positive. False negative (FN) is when the actual class is positive but the model predicts negative.

The ROC curve graphically represents the relationship between sensitivity and specificity. It allows the performance of a model to be evaluated across various classification thresholds. Below are the formulas for sensitivity and specificity.

$$\text{Sensitivity (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{Specificity (TNR)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{8}$$

The AUC represents the area under the ROC curve. The closer the AUC value is to 1, the better the model performance. If the AUC value is close to 0.5, the performance of the model is considered to be at the level of random guessing.

5. Experimental Result

We used accuracy and AUC as the evaluation metrics for performance comparison between our model and the models in [6,18,29,43–46] that have shown excellent performance in various violent-situation detection in recent studies. In datasets like Hockey, Movie-Fights, and Crowd, where the ratio of abnormal to normal data is evenly distributed, model performance was evaluated based on accuracy (ACC).

On the other hand, in the case of datasets with a large imbalance between normal and abnormal data, such as UBI-FIGHT, it is difficult to evaluate model performance using only conventional ACC. Since models can be biased towards the predominant class in imbalanced data, we adopted the AUC evaluation metric, which is widely used to assess performance on imbalanced datasets.

Deep networks, capable of extracting features across various layers, have demonstrated powerful performance on high-dimensional data like videos, but issues such as overfitting can occur, and depth does not always correlate with performance. We experimented with various depths of the backbone model, ResNet-3D, and conducted performance evaluations. Four versions of the widely used classification model proposed (ResNet-10, ResNet-34), were applied in the experiments. By using models of various depths, we sought to explore the relationship between the depth of each model and its performance.

The results in Table 1 are from an experiment using the UBI-FIGHT dataset. For evaluation, the model proposed by Hasan [6], which achieved high performance in the field of violent-situation detection; the model proposed by Ravanbakhsh [43]; Wang's model [44]; the Binary SVM Classifier proposed by Sultani [45]; and the GMM model [47], which exhibits excellent performance in video violence-situation recognition, were compared. Before training and evaluating, the data were resized to 512×512, which is a power of two, ensuring efficient operation in GPU memory allocation and thread management. Since the number of frames in each video within the video dataset can vary, 150 frames were extracted from the frames of each video to train the model. The UBI-FIGHT dataset comprised 784 normal videos and 214 abnormal videos and included a vast array of video lengths ranging from a minimum of one minute to a maximum of one hour. However, due to the significant imbalance between the number of normal and abnormal videos, the AUC metric, which is more suitable for imbalanced datasets, was used instead of the commonly used accuracy (ACC) metric to evaluate the performance of the model. The model based on the proposed ResNet-34 showed outstanding performance, with an AUC score of 0.952.

Table 1. Performance comparison of proposed model and existing models on the UBI-FIGHT dataset.

Method	AUC
Hasan et al. [6]	0.510
Ravanbakhsh et al. [43]	0.523
Wang et al. [44]	0.610
Sultani et al. [45]	0.892
GMM [47]	0.906
Proposed (ResNet-10)	0.951
Proposed (ResNet-34)	0.952

Table 2 presents the experimental results for the Hockey, Movie-Fight, and Crowd datasets. The models used for comparison, which are I3D, Flow Gated, and SPIL, are all networks that have shown good performance in violence-situation detection. Since all the data used in Table 2 are balanced datasets with equal proportions of normal to abnormal instances, accuracy (ACC) was used as the evaluation metric for analysis. In the Hockey dataset, the model proposed in this paper, Proposed (ResNet-34), achieved high performance with an ACC of 98.1. In the Movie-Fights dataset, all depths of ResNet models reached high performance with an ACC of 100. For the Crowd dataset, which involves violent situations occurring within large crowds, the Proposed (ResNet-10) model showed good results with an ACC of 94.0, close to the performance of SPIL.

Table 2. Performance comparison of proposed model and existing models on the Hockey, Movie-Fights, and Crowd datasets.

Method	Hockey (ACC)	Movie-Fights (ACC)	Crowd (ACC)
I3D [6]	0.934	0.958	0.834
Flow Gated	0.980	0.973	0.888
SPIL [45]	0.968	0.953	0.954
Proposed (ResNet-10)	0.970	1.0	0.940
Proposed (ResNet-34)	0.981	1.0	0.915

Table 3 presents a comparative analysis of performance with and without the use of an attention module. The results indicate a higher ACC when the attention module is employed. It demonstrates that assigning weights to significant frames among RGB and optical-flow frames enhances the effectiveness of video violence detection. This finding confirms the value of focusing on crucial frames in improving the recognition of violent situations in videos.

Table 3. Comparative performance evaluation of models with and without attention module.

Method	Hockey (ACC)
Proposed (ResNet-10)	0.970
Proposed (ResNet-10) Without Attention	0.930
Proposed (ResNet-34)	0.981
Proposed (ResNet-34) Without Attention	0.925

Additionally, in terms of time efficiency, the proposed model demonstrated reliable performance with a processing speed of 30–60 frames per second (FPS), aligning well with the operational standards of real-time CCTV systems. Although this aspect was not directly compared with other models, it adds a significant dimension to the practical applicability of our approach in real-world scenarios.

6. Conclusions and Future Work

In this study, we propose an enhanced violence-detection model that integrates optical flow and RGB data by augmenting the ResNet-3D model with an attention module. This model effectively captures both the spatiotemporal features of the dynamic context from optical flow and the spatial context of RGB frames, combining them for a comprehensive analysis. The introduction of the attention module significantly improved the model's ability to focus on crucial frames for detecting violent situations within the combined RGB and optical-flow frames, thus increasing the accuracy of violence detection.

While the foundational ResNet-based ResNet-3D model, which leverages the residual connections of ResNet, allows for efficient training with high-dimensional data such as videos. Performance evaluations on real-life CCTV datasets, including the Crowd, Hockey, UBI-FIGHT, and Movie-Fights datasets, have shown the superior accuracy of our model over existing models in various environments with differing backgrounds, lighting, and

movements. However, the model's performance slightly decreases in scenarios involving videos with a high density of people. Additionally, while the model is capable of processing video in real-time, reducing the memory required to compute optical flow from images remains a challenge for us.

Our future research plans include two primary directions. Firstly, we aim to conduct research on reducing memory usage for optical flow extraction. This is a crucial step to enhance the efficiency of video processing and increase the applicability of real-time systems. Specifically, by improving memory efficiency, we intend to make optical-flow-based models more versatile across a wider range of equipment and environments.

Secondly, we plan to focus on increasing the accuracy of violence detection in environments with spatial constraints, like crowded areas. Such settings pose a unique challenge to violence detection models due to the difficulty in detecting dynamic movements. To address this, we will work on developing more sophisticated data processing techniques and algorithms to optimize the model for high accuracy in complex environments.

Our findings demonstrate that by utilizing optical flow and the attention module, the model is able to focus more on the dynamic aspects of objects, showing robust performance even in video datasets with diverse backgrounds. This video violence detection model is expected to play a significant role in real-time security and surveillance systems in the future, especially in enhancing safety in complex urban environments or large public spaces. Furthermore, the development of this model expands the possibilities of AI-based security systems and paves the way for the development of more sophisticated surveillance technologies. This represents a significant advancement in improving public safety, offering a swift and efficient mechanism for responding to hazardous situations.

Author Contributions: Conceptualization, J.-H.P.; Methodology, J.-H.P.; Software, J.-H.P.; Validation, J.-H.P., M.M. and H.-S.K.; Formal analysis, J.-H.P.; Investigation, H.-S.K.; Resources, H.-S.K.; Data curation, J.-H.P.; Writing—original draft, J.-H.P. and M.M.; Writing—review and editing, J.-H.P., M.M. and H.-S.K.; Visualization, J.-H.P., M.M. and H.-S.K.; Supervision, H.-S.K.; Project administration, H.-S.K.; Funding acquisition, H.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Technology development Program of MSS [RS-2023-00217732] and supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2024-2020-0-01462) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 1999; IEEE: Piscataway, NJ, USA, 1999; Volume 2, pp. 246–252.
2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
3. Laptev, I. On space-time interest points. *Int. J. Comput. Vis.* **2005**, *64*, 107–123. [CrossRef]
4. Hassner, T.; Itcher, Y.; Kliper-Gross, O. Violent flows: Real-time detection of violent crowd behavior. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–6.
5. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. In *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*; Springer: Cham, Switzerland, 2023; pp. 353–374.
6. Hasan, M.; Choi, J.; Neumann, J.; Roy-Chowdhury, A.K.; Davis, L.S. Learning temporal regularity in video sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 733–742.
7. Hinami, R.; Mei, T.; Satoh, S. Joint detection and recounting of abnormal events by learning deep generic knowledge. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3619–3627.
8. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
10. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
12. Mahmoud, M.; Kasem, M.; Abdallah, A.; Kang, H.S. Ae-lstm: Autoencoder with lstm-based intrusion detection in iot. In Proceedings of the 2022 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 26–28 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
13. Mahmoud, M.; Kang, H.S. GANMasker: A Two-Stage Generative Adversarial Network for High-Quality Face Mask Removal. *Sensors* **2023**, *23*, 7094. [CrossRef] [PubMed]
14. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
15. Ranjan, A.; Black, M.J. Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4161–4170.
16. Wulff, J.; Sevilla-Lara, L.; Black, M.J. Optical flow in mostly rigid scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4671–4680.
17. Huang, Z.; Shi, X.; Zhang, C.; Wang, Q.; Cheung, K.C.; Qin, H.; Dai, J.; Li, H. Flowformer: A transformer architecture for optical flow. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 668–685.
18. Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
19. Zhang, S.; Wang, T.; Wang, C.; Wang, Y.; Shan, G.; Snoussi, H. Video object detection base on rgb and optical flow analysis. In Proceedings of the 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), Xi'an, China, 21–22 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 280–284.
20. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
21. Girdhar, R.; Ramanan, D.; Gupta, A.; Sivic, J.; Russell, B. Actionvlad: Learning spatio-temporal aggregation for action classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 971–980.
22. Zhu, Y.; Lan, Z.; Newsam, S.; Hauptmann, A. Hidden two-stream convolutional networks for action recognition. In Proceedings of the Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Revised Selected Papers, Part III 14; Springer: Cham, Switzerland, 2019, pp. 363–378.
23. Sun, S.; Kuang, Z.; Sheng, L.; Ouyang, W.; Zhang, W. Optical flow guided feature: A fast and robust motion representation for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1390–1399.
24. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 20–36.
25. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal residual networks for video action recognition. corr abs/1611.02155 (2016). *arXiv* **2016**, arXiv:1611.02155.

26. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef] [PubMed]
27. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
28. Morais, R.; Le, V.; Tran, T.; Saha, B.; Mansour, M.; Venkatesh, S. Learning regularity in skeleton trajectories for anomaly detection in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 16 2019; pp. 11996–12004.
29. Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; Tao, D. Gmflow: Learning optical flow via global matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8121–8130.
30. Butler, D.J.; Wulff, J.; Stanley, G.B.; Black, M.J. A naturalistic open source movie for optical flow evaluation. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Proceedings, Part VI 12; Springer: Cham, Switzerland, 2012; pp. 611–625.
31. Fritsch, J.; Kuehnl, T.; Geiger, A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In Proceedings of the International Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, 6–9 October 2013.
32. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
33. Degardin, B.; Proença, H. Iterative weak/self-supervised classification framework for abnormal events detection. *Pattern Recognit. Lett.* **2021**, *145*, 50–57. [CrossRef]
34. Mukherjee, S.; Saini, R.; Kumar, P.; Roy, P.P.; Dogra, D.P.; Kim, B.G. Fight detection in hockey videos using deep network. *J. Multimed. Inf. Syst.* **2017**, *4*, 225–232.
35. Khan, S.U.; Haq, I.U.; Rho, S.; Baik, S.W.; Lee, M.Y. Cover the violence: A novel Deep-Learning-Based approach towards violence-detection in movies. *Appl. Sci.* **2019**, *9*, 4963. [CrossRef]
36. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
37. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
39. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. *arXiv* **2018**, arXiv:1805.09501.
40. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
41. Cortes, C.; Mohri, M.; Rostamizadeh, A. L2 regularization for learning kernels. *arXiv* **2012**, arXiv:1205.2653.
42. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
43. Ravanbakhsh, M.; Nabi, M.; Sangineto, E.; Marcenaro, L.; Regazzoni, C.; Sebe, N. Abnormal event detection in videos using generative adversarial nets. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1577–1581.
44. Wang, T.; Qiao, M.; Lin, Z.; Li, C.; Snoussi, H.; Liu, Z.; Choi, C. Generative neural networks for anomaly detection in crowded scenes. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1390–1399. [CrossRef]
45. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6479–6488.
46. Su, Y.; Lin, G.; Zhu, J.; Wu, Q. Human interaction learning on 3d skeleton point clouds for video violence recognition. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part IV 16; Springer: Cham, Switzerland, 2020; pp. 74–90.
47. Degardin, B.M. Weakly and Partially Supervised Learning Frameworks for Anomaly Detection. Ph.D. Thesis, Universidade da Beira Interior (Portugal), Covilhã, Portugal, 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Multi-Granularity Aggregation with Spatiotemporal Consistency for Video-Based Person Re-Identification

Hean Sung Lee [1], Minjung Kim [1], Sungjun Jang [1], Han Byeol Bae [2] and Sangyoun Lee [1,*]

[1] School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, Republic of Korea; hslee2860@yonsei.ac.kr (H.S.L.); mjkima@yonsei.ac.kr (M.K.); jeu2250@yonsei.ac.kr (S.J.)

[2] School of Computer Science and Engineering, Kunsan National University, 558 Daehak-ro, Gunsan-si 54150, Republic of Korea; hbbae@kunsan.ac.kr

* Correspondence: syleee@yonsei.ac.kr; Tel.: +82-2-2123-5768

Abstract: Video-based person re-identification (ReID) aims to exploit relevant features from spatial and temporal knowledge. Widely used methods include the part- and attention-based approaches for suppressing irrelevant spatial–temporal features. However, it is still challenging to overcome inconsistencies across video frames due to occlusion and imperfect detection. These mismatches make temporal processing ineffective and create an imbalance of crucial spatial information. To address these problems, we propose the Spatiotemporal Multi-Granularity Aggregation (ST-MGA) method, which is specifically designed to accumulate relevant features with spatiotemporally consistent cues. The proposed framework consists of three main stages: extraction, which extracts spatiotemporally consistent partial information; augmentation, which augments the partial information with different granularity levels; and aggregation, which effectively aggregates the augmented spatiotemporal information. We first introduce the consistent part-attention (CPA) module, which extracts spatiotemporally consistent and well-aligned attentive parts. Sub-parts derived from CPA provide temporally consistent semantic information, solving misalignment problems in videos due to occlusion or inaccurate detection, and maximize the efficiency of aggregation through uniform partial information. To enhance the diversity of spatial and temporal cues, we introduce the Multi-Attention Part Augmentation (MA-PA) block, which incorporates fine parts at various granular levels, and the Long-/Short-term Temporal Augmentation (LS-TA) block, designed to capture both long- and short-term temporal relations. Using densely separated part cues, ST-MGA fully exploits and aggregates the spatiotemporal multi-granular patterns by comparing relations between parts and scales. In the experiments, the proposed ST-MGA renders state-of-the-art performance on several video-based ReID benchmarks (i.e., MARS, DukeMTMC-VideoReID, and LS-VID).

Keywords: video-based person re-identification; spatiotemporal learning; attention mechanism; complementary learning

1. Introduction

Person re-identification (ReID) is an essential application in large-scale surveillance systems and smart cities, aiming to identify individuals across different times and locations amidst varying conditions (e.g., camera views, occlusion, background clutter, illumination, scale, and body pose). With the growth of video surveillance systems, advanced video ReID methods [1–8] have been attracting attention due to their potential to offer larger capacity for achieving more robust performance. In contrast to image-based ReID, which relies solely on a single image, video-based approaches harness a richer source of temporal information. Consequently, most video-based methods [6–14] predominantly concentrate on feature extraction and aggregating such spatiotemporal knowledge.

Previous methods commonly fall into two categories: part-based [14–17] and attention-based approaches [4,7,11,13,18–20]. These methods segment global features into partial

information and then derive a single feature vector by leveraging relationships among relevant spatial and temporal knowledge. Specifically, the methods [14,15] spatially separate images or global features into fixed partitions. Ref. [14] employed horizontal separation and utilized the graph convolutional network (GCN) [21] to aggregate spatial and temporal dimensions. Similarly, Ref. [15] employed horizontal separation at multiple scales to divide details and aggregate them through hypergraphs at various granularity levels. Despite these efforts, challenges persist due to temporal misalignment caused by object occlusion or inaccurate detection algorithms during feature aggregation. When features are separated into horizontal parts, they inconsistently include unnecessary information along the temporal axis during occlusion or detection errors (Figure 1a). Such inconsistencies, particularly in video ReID, potentially lead to interference in features and result in inaccurate outcomes. Alternatively, attention mechanisms such as [22–25] have been widely utilized to enhance feature representation by accentuating relevant regions while suppressing irrelevant areas. Recent video-based ReID approaches [18,19] have explored attention-based partitioning methods to leverage diverse attention parts. However, these methods typically create sub-attentions separate from the fixed main attention, resulting in imbalanced information across parts and restricting the number of semantic part divisions. They may tend to prioritize parts with abundant information, potentially overlooking finer details of targets with relatively lesser information. This could result in inaccurate outcomes when crucial parts of the target are occluded. To overcome the above problem, we aim to extract enhanced ReID features by fully exploiting detailed information in the spatiotemporal information by ensuring uniform information quantity across parts and maintaining consistent semantic information temporally.

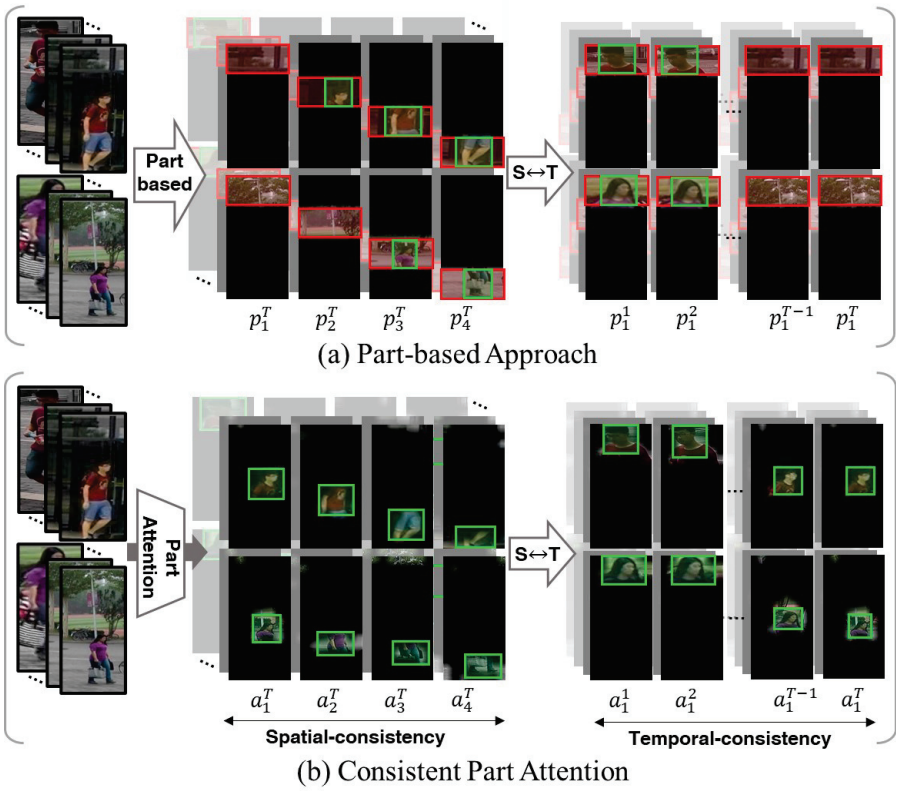


Figure 1. Comparison of the (**a**) part-based approach and (**b**) proposed consistent part-attention (CPA) method. As shown in the figure, the part-based approach has a different amount of relevant spatial information with some interferences for each part, whereas the CPA method provides only key information uniformly. When temporal misalignment occurs, the part-based approach demonstrates inconsistent human parts. In contrast, CPA consistently offers semantic information about the same part.

In this paper, we introduce the *consistent part-attention* (CPA) module, which effectively manages uniform spatiotemporal information without interference or noise. Notably, CPA learns uniform attention in spatiotemporal dimensions solely through self-information and a few priors, eliminating the need for hard labels such as human parsing or skeleton data. As illustrated in Figure 1b, the CPA module not only eliminates interference and noise in spacetime, but also ensures consistent delivery of semantic information to the model, averting uneven information distribution and ensuring the thorough capture of fine target details.

Addressing the challenge of video ReID entails leveraging both spatial and temporal knowledge across various granularities. To this end, we employed the *Multi-Attentive Part Augmentation* (MA-PA) scheme to obtain multi-granularity information with various attention scales. Multi-granularity information [15,26] has shown promise in incorporating detailed features in videos. The MA-PA generates multi-granularity attention by recombining fine attention from CPA. As shown in Figure 2a, merging segmented part information can alter and diversify semantic meanings (i.e., when the target is small, distinguishing facial or footwear details at a smaller scale becomes challenging; however, combining these with parts related to shirts or pants extends semantic information to upper and lower body regions). This empowers the model to capture robust information across a spectrum of semantic meanings, from fine-grained to broader details.

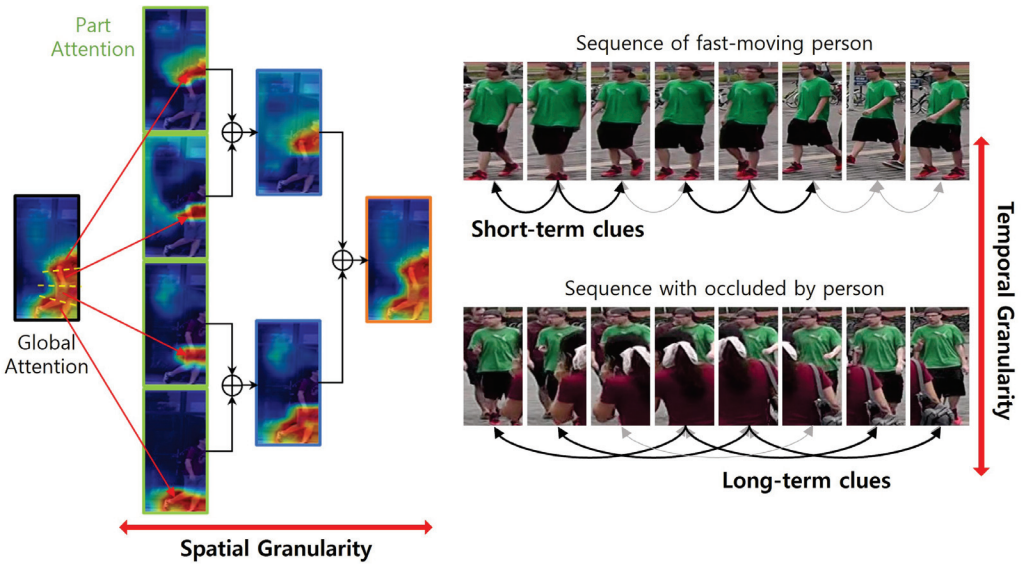


Figure 2. Illustration of spatiotemporal multi-granularity. (**a**) Part-attention-based spatial granularity synthesizes diverse semantic attention at multiple granular levels by combining sub-part attentions. (**b**) Long-/short-term-based temporal granularity comprises granularities with varying sampling intervals to adopt the context of different sequences for the long and short terms.

To capture temporal relations, we employed the *Long-/Short-term Temporal-Augmentation* (LS-TA) module, which obtains multi-granularity temporal information. LS-TA conducts time sampling at different intervals to harness the overall temporal advantage. Long- and short-term temporal cues have been utilized for temporal modeling due to their respective crucial patterns [19,27,28]. As shown in Figure 2b, varying sampling intervals yield distinct features. For instance, short-term clues reflect the target's motion patterns, while long-term cues effectively alleviate occlusion. Consequently, LS-TA yields diverse temporal features, enabling the model to extract robust features in various situations. After augmenting spatial and temporal granular cues, we propose the *Spatiotemporal Multi-Granularity Aggregation* (ST-MGA) to exploit densely separated spatial and temporal clues simultaneously. ST-MGA

investigates the relations between multi-granular and part information from both spatial and temporal cues. Since the granular part features refined in the previous process contain all the information in spatiotemporal dimensions without interference, ST-MGA can extract robust and complementary features in any situation.

To summarize, our main contributions are as follows. We designed a *consistent part-attention* (CPA) module to provide spatiotemporally consistent and well-aligned part attention. To exploit the multi-scale granularity, we introduce *Multi-Attention Part Augmentation* (MA-PA), which uses the fine part features from CPA to synthesize semantic parts at multiple scales spatially. We also suggest *Long-/Short-term Temporal Augmentation* (LS-TA) for considering relations at various temporal scales. The temporally consistent part information through CPA allows LS-TA to have the full advantage of temporal knowledge. Using the spatiotemporal multi-granularity part information, we propose *Spatiotemporal Multi-Granularity Aggregation* (ST-MGA), which performs partwise and scalewise aggregation. The ST-MGA method investigates the relations between multi-granular and part information from both spatial and temporal cues and encourages complementary features for video person ReID. In the experiments, we validated the effectiveness of our approach on multiple benchmarks. Our approach outperforms previous state-of-the-art methods on several video ReID benchmarks and shows more accurate attention parts than the existing part-based approaches.

2. Related Work

2.1. Video-Based Person ReID

In recent years, video-based person ReID [7,14,15,17–20,26–43] has garnered significant attention due to the abundant temporal and spatial cues available in videos. The predominant approach in video ReID is extracting and aggregating dynamic spatiotemporal features. Some methods employ recurrent architectures [5,6,44] for video representation learning to leverage temporal cues. Refs. [28,45] utilized 3D-convolution [46,47] for spatial–temporal feature learning. A temporal attention mechanism [8,9,20,48] has also been proposed for robust temporal feature aggregation. In recent research, to contain richer temporal and spatial information, many methods [20,39–42] have been proposed. Ref. [39] presented a statistic attention (SA) block to capture long-range high-order dependencies of the feature maps. Ref. [40] used hierarchical mining, which mines the characteristics of pedestrians by referring to the temporal and intra-class knowledge. Ref. [41] proposed a saliency and granularity mining network to learn the temporally invariant features. Ref. [42] implemented a two-branch architecture to separately learn the pose feature and appearance feature and concatenated them together for more discriminative representation. Ref. [20] removed interference and obtained key pixels and frames by learning attention-guided interference-removal modules. A simple literature survey of the previous methods is shown in Table 1.

Recently, refs. [7,26] focused on aggregating diverse partial information, both spatially and temporally. To obtain partial spatial cues, certain approaches have adopted horizontal partitioning [14–17] or explored diverse attention mechanism [7,11,18,19]. However, most of these methods cannot fully exploit the potential of spatiotemporal knowledge. Horizontal partitioning often struggles to maintain information consistency in cases of temporal misalignment due to an inaccurate detector. The diverse attention mechanisms have an unbalanced information distribution regarding attention, leading to inefficient aggregation. To exploit the full advantage of spatiotemporal information, we first propose a straightforward, yet effective framework, called *consistent part attention* (CPA), designed to ensure the consistency of partial information and lead to efficient aggregation in the spatial and temporal dimensions. Then, we efficiently the aggregate spatiotemporal partial information using the *Spatiotemporal Multi-Granularity Aggregation* (ST-MGA) scheme to extract complementary video features.

Table 1. Methodology of video-based person re-identification.

Reference	Source	Methodology
M3D [28]	AAAI'19	Proposed multi-scale 3D-convolution layer to refine the temporal features
STA [7]	AAAI'19	Proposed spatial–temporal attention approach to fully exploit discriminative parts
GLTR [27]	ICCV'19	Proposed global–local temporal representation to exploit multi-scale temporal cues
RGSA [29]	AAAI'20	Designed relation-guided spatial-attention module to explore discriminative regions
FGRA [30]	AAAI'20	Proposed frame-guided region-aligned model to extract well-aligned part features
MG-RAFA [26]	CVPR'20	Suggested attentive feature aggregation with multi-granularity information
PhD [31]	CVPR'20	Proposed Pompeiu–Hausdorff distance learning to alleviate the data noise problem
STGCN [14]	CVPR'20	Jointly optimized two GCN branches in spatial and temporal dimensions for complementary information
MGH [15]	CVPR'20	Designed a multi-granular hypergraph structure to increase representational capacities
TCLNet [18]	ECCV'20	Introduced a temporal-saliency-erasing module to focus on diverse part information
AP3D [32]	ECCV'20	Proposed appearance-preserving 3D-convolution to align the adjacent features at the pixel level
AFA [33]	ECCV'20	Proposed adversarial feature augmentation, which highlights the temporal coherence features
SSN3D [34]	AAAI'21	Designed a self-separated network to seek out the same parts in different frames
BiCnet-TKS [19]	CVPR'21	Used multiple parallel and diverse attention modules to discover diverse body parts
STMN [35]	ICCV'21	Leveraged spatial and temporal memories to refine frame-/sequence-level representations
STRF [36]	ICCV'21	Proposed spatiotemporal representation factorization for learning discriminative features
SINet [37]	CVPR'22	Designed SINet to enlarge attention regions for consecutive frames gradually
CAVIT [38]	ECCV'22	Used a contextual alignment vision transformer for spatiotemporal interaction
SANet [39]	TCSVT'22	Introduced the SA block, which can capture long-range and high-order information
HMN [40]	TCSVT'22	Designed hierarchical mining network which can mine as many characteristics
SGMN [41]	TCSVT'22	Designed a saliency- and granularity-mining network for discovering temporal coherence
BIC+LGCN [42]	TCSVT'23	Used a branch architecture to separately learn appearance features and human pose
IRNet-V [20]	TCSVT'23	Proposed an interference-removal framework for removing various interferences

2.2. Attention for Person ReID

The attention mechanism, as discussed in [22–25], has been widely used in person ReID to enhance representation by emphasizing the relevant features and suppressing irrelevant ones. In image-based ReID, Refs. [49–58] learned attention in terms of the spatial or channel dimension. In some studies of video-based ReID, temporal attention is performed to weigh and aggregate frame-level features [5,9,59]. Moreover, Ref. [26] proposed joint spatial and temporal attention to exploit relations at multiple granularities. Recently, Refs. [18,19] proposed diverse spatial-attention modules to enhance video representation. The diverse attention modules focus on different regions for consecutive frames. However, they create sub-attention parts separate from the attention of the main frame, which restricts the number of semantic attention parts and results in each part containing an inconsistent amount of information. This limitation leads to inefficiencies in aggregating diverse features because the focus remains on the main attention part. Unlike the above methods, the proposed CPA provides uniform spatial information and temporally coincident cues about the diverse attention parts, leading to efficient aggregation in the spatial and temporal dimensions.

2.3. Spatiotemporal Aggregation

Capturing spatial and temporal information is critical to learning comprehensive representations of videos effectively. The most-used approach [4,18,27,32,34] involves using convolutional neural networks (CNNs) to extract spatial features from individual video frames and integrating these features with temporal modeling. Various methods, such as RNNs, 3D-CNNs, GCNs, and attention mechanisms, can be employed for spatiotemporal aggregation. Ref. [28] introduced a compact 3D convolutional kernel that facilitates multi-scale temporal feature learning by incorporating long-term temporal modeling and refining appearance features through spatiotemporal masking. Ref. [19] focused on capturing visual features by considering the spatial details and long-distance context information, which are combined using a multi-scale temporal kernel in the 3D convolutional layers. Ref. [60] proposed graph convolution to directly propagate cross-spacetime and cross-scale information, capturing high-order spatial–temporal correlations. Ref. [35] addressed the problem of spatial distractors by memorizing them and suppressing distracting scene details while using temporal attention patterns to aggregate the frame-level representation.

Ref. [61] learned spatiotemporal information attention using ConvLSTM [62] to explicitly capture and aggregate spatiotemporal information in video-based industrial smoke emission recognition. Ref. [63] explored spatial correlations within each frame to determine the attention weight of different locations and also considered temporal correlations between adjacent frames.

3. Methodologies

In this section, we propose *Spatiotemporal Multi-Granularity Aggregation* (ST-MGA) methods with the proposed spatiotemporally consistent cues. We introduce the preliminaries in Section 3.1. Then, we describe the proposed *consistent part-attention* (CPA) module, which aims to provide spatiotemporally consistent information by using a simple, but effective attention approach, in Section 3.2. For multi-granular spatial and temporal information, we introduce *Multi-Attention Part Augmentation* (MA-PA) and *Long-/Short-term Temporal Augmentation* in Section 3.3. Last, we present ST-MGA for spatiotemporally complementary features in Section 3.4.

3.1. Overview

To improve video-based person ReID, we aimed to extract consistent spatial and temporal cues, and aggregated this information for complementary feature extraction. The overall framework of the proposed approach is illustrated in Figure 3. As the input video clip, we randomly sampled T frames as $\mathcal{V} = \{I_1, I_2, \cdots, I_T\}$. Then, we extracted video features $F \in \mathbb{R}^{T \times C \times H \times W}$ through the CNN backbone (e.g., ResNet-50 [64] pretrained on ImageNet [65]), where H, W, and C represent the height, width, and number of channels, respectively. To extract the part cues, we employed the CPA module, which ensures that extracted attentions contain consistent spatiotemporal semantic information. Using CPA, we obtained N_s sub-attentions, $A = \{a^n | n = 1, 2, \cdots, N_s\}$. Subsequently, we extracted multi-granular parts using the MA-PA method. MA-PA augments multi-granular attention with varying scales by utilizing attention parts from CPA. Augmented multi-granular attention was applied to F to generate N_P granular features. To consider temporal relations, we also jointly augmented N_t long- and short-term features using LS-TA by combining multi-scale temporal cues. We define part features after MA-PA and LS-TA as $\hat{P}$ and $\tilde{P}$, respectively. Leveraging $N_p \times N_t$ multi-granular features, we propose ST-MGA, which exploits spatial and temporal relations and aggregates different levels of part features. Finally, after partial and temporal averaging of the aggregated features, we extracted the complementary video features, $Z_c \in \mathbb{R}^C$. The notations and their corresponding descriptions are presented in Table 2.

Table 2. Notations and descriptions.

Notations	Descriptions
$\mathcal{V}$	Input video
T	Length of $\mathcal{V}$
F	Output feature from backbone
A	Set of part attentions from CPA
$\dot{A}$	Set of residual part attentions in CPA
P	Set of part-attentive features
$\hat{A}$	Set of augmented part attentions from MA-PA
$\hat{P}$	Set of augmented spatial attentive features after MA-PA
a'	Temporal attention value in LS-TA
$\tilde{P}$	Set of augmented multi-granular features after LS-TA
Z^F	Global averaged feature from backbone
Z^T	Temporal aggregated feature from T-MGA
Z^C	Final complementary video feature from S-MGA

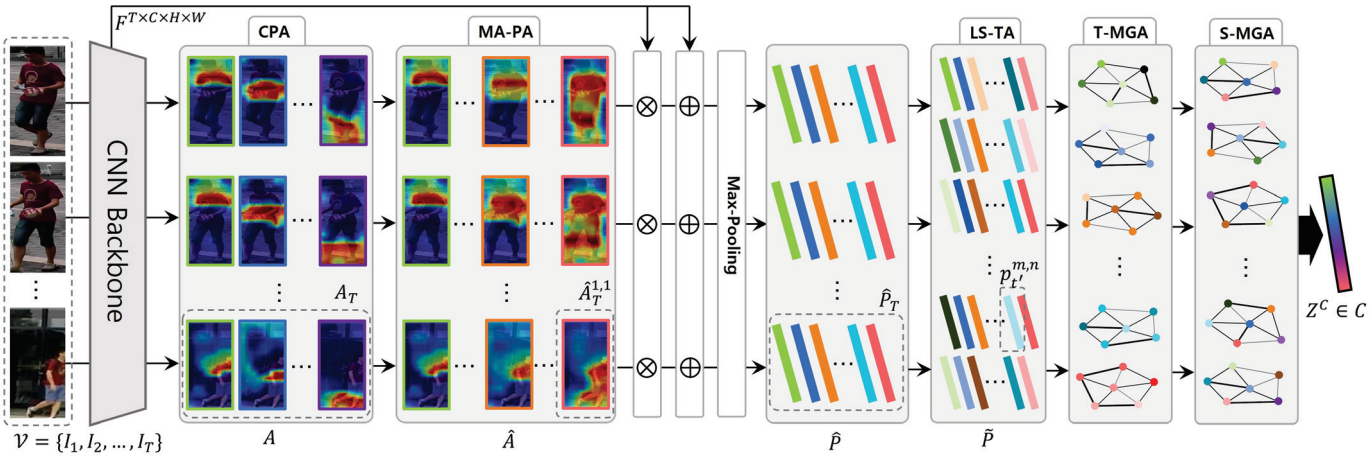


Figure 3. Overview of our framework. We sampled T frames in the video sequence and extracted features F from the CNN backbone. We first extracted sub-attentions A through the CPA module with spatiotemporally consistent information. For multiple spatial granularities, we augmented part attention $\hat{A}$ by combing A hierarchically. Subsequently, we extracted the part-attentive features $\hat{P}$ by applying $\hat{A}$ and the max-pooling operation to F. We jointly augmented $\hat{P}$ to $\tilde{P}$, which contain long-/short-term features, to consider temporal relations. Leveraging the augmented-spatiotemporal-part features, we exploited all video cues by aggregating the part features spatially (S-MGA) and temporally (T-MGA). Last, we extracted the complementary video person ReID features.

3.2. Spatiotemporally Consistent Part Attention

Given the t-th global features $F_t \in \mathbb{R}^{H\times W\times C}$, we first optimized the global attention through a learnable model. We extracted the global attention value a_t^g for the t-th frame from the global-attention module GA composed as follows:

$$a_t^g = Sigmoid(W_2 ReLU(W_1 F_t)), \tag{1}$$

where $W_1 \in \mathbb{R}^{\frac{C}{\gamma}\times C}$ and $W_2 \in \mathbb{R}^{1\times\frac{C}{\gamma}}$ are implemented by 1×1 convolution with shrink ratio γ followed by BN. Subsequently, we can extract global attentive features $P_g \in \mathbb{R}^C$ as:

$$P_g = \frac{1}{T}\sum_{t=1}^{T}(\mathcal{P}(F_t + a_t^g \odot F_t)), \tag{2}$$

where the symbol $\odot$ represents elementwise multiplication and $\mathcal{P}$ is the global max-pooling operation. To optimize a^g, we applied the batch hard triplet loss [66] and softmax cross-entropy loss with P_g as the input. The two loss formulas are denoted as $\mathcal{L}_{tri}^g$ and $\mathcal{L}_{CE}^g$.

After obtaining a^g, we optimized the CPA by using a^g. As illustrated in Figure 4, CPA composes N_s part-attention modules PA, which extract N_s different semantic sub-attentions $A_t = \{a_t^n\}_{n=1}^{N_s}$. Each PA module has the same structure as GA, with different parameters. To encourage each attention focus to have different semantic information, we define the following priors: (i) the features applied by each attention must be semantically classifiable for each other; (ii) the sum of all sub-attention parts should be global attention; (iii) each sub-attention part should contain a uniform amount of information.

Based on the first prior, we designed a part classifier that classified N_s parts as the class ID. We first extracted the part-attention features $P = \{P_n\}_{n=1}^{N_s}$ in the same way as P_g with the corresponding attention. Then, we trained each attentive feature to be classifiable by setting a different label y^n, a one-hot vector for the n-th attentive part. To this end,

we employed the softmax cross-entropy loss with input P_n. We define this loss formula as follows:

$$\mathcal{L}_{PCE} = \mathbb{E}[-log(p(y^n|P_n))], \tag{3}$$

where $p(y^n|P_n)$ is the predicted probability that input P_n belongs to its part label y^n. $\mathcal{L}_{PCE}$ encourages each attention to focus on different relevant areas because each attentive feature must be semantically classifiable despite being applied to the same input features.

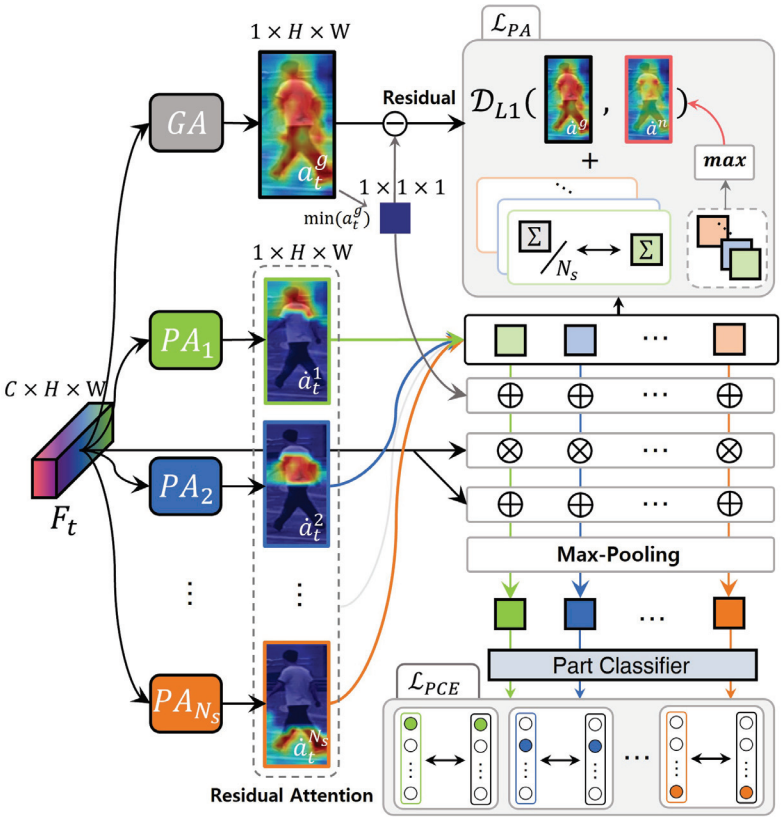


Figure 4. Learning framework for the consistent part-attention (CPA) module. The CPA includes one global-attention module, GA, to extract global attention a^g, N_s part-attention modules, PA, and one part classifier, C_P. In CPA, we used a^g to encourage each attention part to be optimized based on predefined priors.

Before applying the second and third priors, we introduced a technique to extract precise part-attention information. The technique is that sub-attentions only learn the residual from the background value of the global attention when directly inducing each attention part. The rationale behind this approach is to focus on the relevant region only, rather than the background. If background information is not excluded, arbitrary attention is slightly highlighted on the background region as the interest, making interference during aggregation. The goal is for each attention part to have equal importance; thus, we proceeded with learning by extracting only the residuals from the background values of the global attention. By excluding the background information, attention is directed to the relevant areas only. We analyze this in Section 4.

As shown in Figure 4, we first define the minimum of a^g of the background score and obtained the residual global attention, $\dot{a}^g$, by removing it. We define the residual part attentions as $\dot{A} = \{\dot{a}^n\}_{n=1}^{N_s}$, which are learned by focusing only on the semantic area and ignoring unnecessary parts (e.g., the background and occlusions). We utilized $\dot{a}$ and $\dot{a}^g$ to impose constraints on the second and third priors. First, we designed the $L1$ distance loss, which sets the sum of all residual part attentions $\dot{A}$ equal to $\dot{a}^g$. Additionally, we

encouraged the spatial sum of each residual part attention to match the spatial sum of $\tilde{a}^g$ divided by N_s. We combined the first and second loss formulas as follows:

$$\mathcal{L}_{PA} = || \sum_{n=1}^{N_s} \dot{a}^n - \dot{a}^g || + \sum_{n=1}^{N_s} || \sum_{k=1}^{K} \dot{a}_k^n - \frac{1}{N_s} \sum_{k=1}^{K} \dot{a}_k^g ||, \tag{4}$$

where $K = H \times W$. $\mathcal{L}_{PA}$ simply, but strongly satisfies the prior we predefined. Furthermore, $\mathcal{L}_{PA}$ encourages an emphasis on the position of relevant and uniform importance in each part.

The overall loss function for training CPA is formulated as follows:

$$\mathcal{L}_{CPA} = \mathcal{L}_{tri}^g + \mathcal{L}_{CE}^g + \mathcal{L}_{PA} + \lambda \mathcal{L}_{PCE}, \tag{5}$$

where λ is a scaling parameter for weighting $\mathcal{L}_{PCE}$. The attention parts extracted using CPA represent temporally consistent semantic cues with uniform spatial importance. We leveraged these attention parts instead of the previous simple horizontal partitions to take full advantage of this in the video. In subsequent sessions, we will denote $\tilde{A}$ as A for convenience.

3.3. Multi-Granularity Feature Augmentation

To further improve video feature representation, we augmented diverse spatial and temporal cues at multi-granular levels. With the efficient CPA module to obtain consistent part attention A, we applied Multi-Attentive Part Augmentation (MA-PA), which uses A as a sub-attention to generate the parent attention. Following [15], we hierarchically combined the sub-attention into a granular scale $m \in \{0, 1, 2, \cdots, M\}$. To avoid excessive duplication of certain parts, we prevented overlapping when combining sub-attentions, as shown in Figure 2a. The parent granular attention comprises the non-duplicated combination of child sub-attentions. We define the t-th augmented part attention as $\hat{A}_t = \{A_t^m | m = 0, 1, 2, \cdots, M\}$, where $\hat{A}_t^m = \{\hat{a}_t^{m,n} | n = 1, 2, \cdots, 2^m\}$. $\hat{a}_t^{m,n}$ represents attention to the n-th part of the m-th scale, and each scale has 2^m part attentions. Subsequently, we obtained $N_P = \sum_{m=0}^{M} 2^m$ attentive features $\hat{P}$ in each frame as follows:

$$\hat{p}_t^{m,n} = \mathcal{P}(F_t + \hat{a}_t^{m,n} \odot F_t), \tag{6}$$

where $\hat{p}_t^{m,n} / \hat{a}_t^{m,n}$ is the n-th part feature/attention with scale m and $\mathcal{P}$ is the global max-pooling operation. As with $\hat{P}$, each part feature at the same granular level contains uniform semantic information.

Using augmented spatial attentive features $\hat{P}$, we employed Long-/Short-term Temporal Augmentation (LS-TA) to augment long- and short-term features jointly. As shown in Figure 2b, the importance of long- and short-term temporal relations can vary depending on the sequence context. However, simple temporal processing is often inefficient due to temporal inconsistencies such as misalignment between adjacent frames [32]. In our study, we have already addressed these temporal inconsistency problems with the proposed CPA; therefore, we can proceed with the temporal granularity aggregation without concern. To this end, we first applied temporal attention from each frame level by predicting the framewise scores as follows:

$$\tilde{p}_t = \hat{p}_t + a'_t \odot \hat{p}_t, \tag{7}$$

$$a'_t = Sigmoid(W'_2 ReLU(W'_1 hayp_t)), \tag{8}$$

where a'_t is the temporal attention value, $W'_1 \in \mathbb{R}^{\frac{C}{\gamma} \times C}$ and $W'_2 \in \mathbb{R}^{1 \times \frac{C}{\gamma}}$ are implemented by 1D-convolution with a kernel size of 1, and γ is the shrink ratio. Then, following [19], we conducted a temporal-select operation $\{\mathcal{S}^{t'} : \hat{P} \rightarrow \tilde{P}_{t'} \in \mathbb{R}^{T \times C}\}$, where $\mathcal{S}$ is the frame-select operation of a 1D-temporal convolution [67] with a kernel size of $2t' + 1$ and $t' \in 1, 2, \cdots, T'$

is the temporal granular scale. We varied t' to obtain multiple temporal granular features at different intervals from short to long term. With LS-TA, we generated $N_T = T \times T'$ temporal granular features in each part and combined them with the granular part, and a total of $N_P \times N_T$ spatiotemporal granular cues were stored. We leveraged these granular cues to exploit the spatiotemporal coverage in the video.

3.4. Spatiotemporal Multi-Granularity Aggregation

To extract optimal video person ReID features, it is important to effectively aggregate the multi-granular information from the spatial and temporal information. From this view, we propose a Multi-Granularity-Aggregation (MGA) module that fully exploits the spatiotemporal cues. The structure of MGA is illustrated in Figure 5. MGA primarily comprises partwise and scalewise aggregators. The partwise aggregator has M parallel branches with the same structure, where M is the simplified scale number. For each scale of the granularity level, we represent N part features as $\tilde{p}_i^m \in \mathbb{R}^{N \times C}$, where $i = \{1, \cdots, N\}$ and N is the simplified number of parts. The partwise relation matrix in the m-th scale $R^m \in \mathbb{R}^{N \times N}$ can be defined as a dot-product affinity as follows:

$$R_{i,j}^m = \frac{\exp(\varphi(\tilde{p}_i^m)^T \omega(\tilde{p}_j^m)/\tau)}{\sum_{k=1}^{N} \exp(\varphi(\tilde{p}_i^m)^T \omega(\tilde{p}_k^m)/\tau)}, \tag{9}$$

where φ and ω are linear functions and τ is the temperature hyperparameter. With the relation matrix R^m, we calculated partwise aggregated features as $\phi(\tilde{p})^T R_m$, where ϕ is a linear function. Then, the output features were extracted using average pooling of all part cues, followed by the residual addition. After the partwise aggregator, we concatenated M outputs, denoted as $Y \in \mathbb{R}^{M \times C}$.

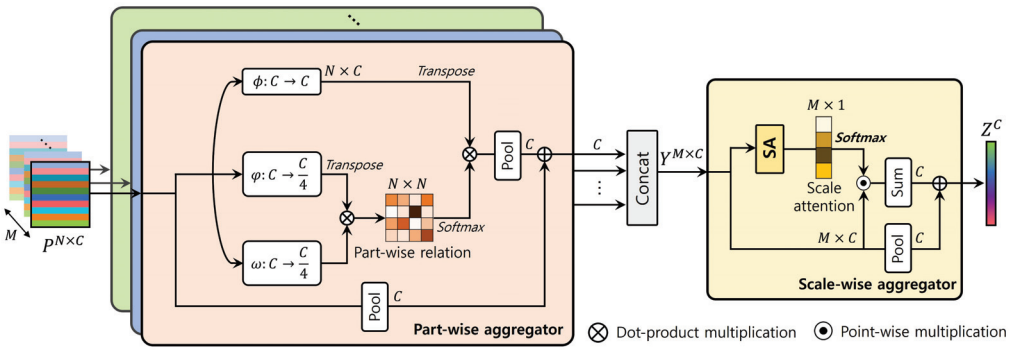


Figure 5. The architecture of Multi-Granularity Aggregation (MGA). As the input $\tilde{P}$, partwise aggregation proceeds first at each granular scale. Then, the granular features X are aggregated using a scale-attention block. Last, MGA extracts complementary video ReID features, Z. Temporally, N and M are changed to T and T', respectively.

With input Y, scalewise aggregation proceeds using a scalewise aggregator. To aggregate all granular scale cues on Y, we designed a scale-attention (SA) module that predicts the scalewise attention score to weigh the aggregation. We normalized the learned attention scores from the SA module via the softmax function across scale dimensions and obtained the scale-attention score $A_s \in \mathbb{R}^{C \times M}$:

$$A_s = SA(Y) = Softmax(W_{\theta_2}(ReLU(W_{\theta_1} Y))/\tau), \tag{10}$$

where $W_{\theta_1} \in \mathbb{R}^{\frac{C}{\eta} \times C}$ and $W_{\theta_2} \in \mathbb{R}^{1 \times \frac{C}{\eta}}$ are a fully connected layer with shrink ratio η and τ is the temperature hyperparameter. Afterward, we extracted complementary video representation $Z \in \mathbb{R}^C$ using scalewise attention A_s to aggregate all scale features as follows:

$$Z = \sum_{m=1}^{M} A_s \odot Y + \mathcal{A}(Y). \tag{11}$$

We applied MGA separately in the spatial (S-MGA) and temporal (T-MGA) information. S-MGA aggregates partwise and scalewise based on spatially attentive-part features, whereas T-MGA is based on the temporal-part features. As illustrated in Figure 5, T-MGA has the same structure as S-MGA, with N and M replaced by T and T', respectively. By applying T-MGA and S-MGA, we extracted final video features Z^C, enhanced by exploiting the spatiotemporal cues across each scale.

For the reliable learning of each stage, we used three types of features $Z' = \{Z^F, Z^T, Z^C\}$, where Z^F is the global features of the backbone, Z^T represents the temporally aggregated features from T-MGA, and Z^C denotes the final complementary video features from S-MGA. The overall joint objective is defined as follows:

$$\mathcal{L}_{total} = \sum_{z \in Z'} (\mathcal{L}_{CE}^{z} + \mathcal{L}_{tri}^{z}) + \mathcal{L}_{CPA}, \tag{12}$$

where $\mathcal{L}_{CE}$ is the softmax cross-entropy loss and $\mathcal{L}_{tri}$ is the batch-hard triplet loss [66]. In the inference phase, we only used the complementary video person ReID features, Z_C.

4. Experiments

In this section, we extensively evaluate the proposed framework in video-based person ReID scenarios. We conducted comprehensive experiments on three benchmark datasets, and provide a detailed analysis of the results. Additionally, we included extensive ablation studies to investigate the effectiveness of our approach further.

4.1. Datasets and Evaluation Metric

We evaluated the proposed framework on three challenging video ReID datasets: MARS [2], DukeMTMC-VideoReID (Duke-V) [68], and LS-VID [27]. MARS is one of the large-scale benchmark datasets for video ReID, which contains 17,503 tracklets of 1261 identities and an additional distractor of 3248 tracklets. There are substantial bounding box misalignment problems to make it more challenging. Duke-V is a widely used large-scale video ReID dataset captured by 8 cameras with 4832 tracklets of 1404 identities. LS-VID is the most recent large-scale benchmark dataset for video ReID. It contains 3772 identities and 14,943 tracklets captured by 15 cameras. There are many challenging elements, such as varying illumination and bounding box misalignment, to make it close to a real-world environment, A summary comparison is illustrated in Table 3. For the evaluation, we used only the final complemented video ReID features Z_C during the inference stage. Moreover, we assessed the performance using Rank-1, Rank-5, and Rank-10 accuracy for cumulative matching characteristics (CMCs) and the mean average precision (mAP). Rank-k and the mAP are the most popular evaluation metrics for person ReID. Rank-k measures the accuracy by evaluating whether the correct match appears within the top-k-ranked results. To this end, for each query, an algorithm will rank all the gallery samples according to their distances. The mAP evaluates how well the system ranks the retrieved matches for each query. It considers both precision and recall, providing a comprehensive assessment of the re-identification system's performance across different query scenarios.

Table 3. Statistics between different video ReID datasets.

	MARS	**Duke-V**	**LS-VID**
# Identities	1261	1404	3772
# of Videos	20,751	4832	14,943
# of Cameras	6	8	15
B-Box	DPM	manual	FRCNN

4.2. Implementation Details

Following the common practices in ReID [19,27,32,69], we used ResNet-50 [64] trained by ImageNet classification [65] as the backbone of the proposed model for a fair and effectiveness validation. Similar to [69,70], we removed the last down-sampling operation to enrich granularity, resulting in a total down-sampling ratio of 16. We employed a random sampling strategy to select 8 frames for each video sequence with a stride of 4 frames. Each batch contains 16 identities, each with 4 tracklets. The input frames were resized to 256×128. Random horizontal flip and random erasing [71] were used for data augmentation. The training process utilizes the Adam [72] optimizer for 150 epochs. The learning rate was initialized to 3.5×10^{-4} and decayed every 40 epochs using a decay factor of 0.1. For the hyperparameters of the proposed framework, a greedy search was conducted about λ and τ to increase reproducibility and transparency. λ is the weight for $\mathcal{L}_{PCE}$ of Equation (5), and τ is the temperature hyperparameter, which indicates sensitivity to the relation of Equations (9) and (10). As shown in Figure 6, we set λ to 0.1 and τ to 0.05 as the optimal parameters. The shrink ratios γ and η were set to 16 and 4, respectively. The framework was implemented with *PyTorch1.4* [73] on one GeForce RTX 3090 (Nvidia, Santa Clara, CA, USA).

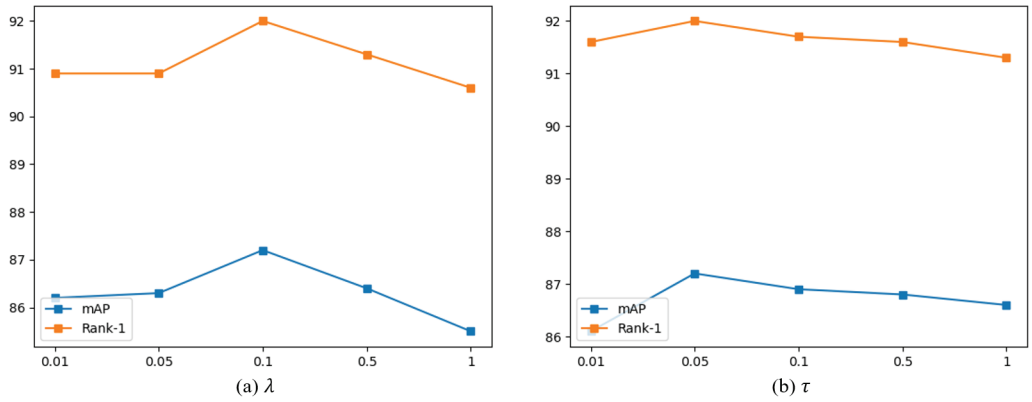


Figure 6. Sensitivity analysis on hyperparameters for (**a**) λ and (**b**) τ in MARS [2].

4.3. Ablation Study

4.3.1. The Influence of CPA

We first evaluated the effect of the proposed CPA module with different components of loss functions and techniques. To extract spatiotemporally consistent information, we suggest three priors and approaches with $\mathcal{L}_{PCE}$, $\mathcal{L}_{PA}$, and a residual technique (*res*). We conducted only spatial aggregation with three granular-part attentions (1, 2, and 4 partitions). $\mathcal{L}_{PCE}$ encourages the attentions from CPA to focus on different sub-areas. In Table 4, employing $\mathcal{L}_{PCE}$ brings 0.3%/0.8% mAP/Rank-1 gains, which is the same as the part-based approach (in the second row in Table 5). Thus, employing $\mathcal{L}_{PCE}$ can only successfully separate the attention parts and lead to aggregation between separate part information. $\mathcal{L}_{PA}$ performs a comparison with the global attention and spatially unifies the amount of information in each attention, which allows separate attentions to focus on the more semantic areas, making all attention parts useful in the aggregation. However, the comparison with the global attention inevitably includes some scores for the background; thus, $\mathcal{L}_{PA}$ alone does not ensure that the attention parts focus on the fully semantic area. To address this problem, learning only the residuals excluding the background scores (*res*) results in a 1.6%/1.8% mAP/Rank-1 performance increment in the last row of Table 4. The improvement proves that CPA is effective.

Table 4. Performance of consistent part-attention (CPA) module in MARS [2] under different loss components. **Bold** denotes best performance.

$\mathcal{L}_{PCE}$	$\mathcal{L}_{PA}$	*res*	mAP	Rank-1	Rank-5	Rank-10
			85.2	89.1	86.7	97.5
✓			85.5	89.9	96.9	97.7
✓	✓		85.6	90.1	96.9	97.9
✓	✓	✓	**86.8**	**90.9**	**97.4**	**98.0**

For a more detailed analysis of CPA, we provide a few contextual comparisons to visualize how the methods within CPA affect the actual attention map. A comparison of the attention map visualizations for CPA is presented in Figure 7. Using only $\mathcal{L}_{PCE}$ to learn to be classifiable for each attention (2nd row) results in partitions separated by a constant amount, like the horizontal partition (1st row). However, it concurrently generates significant unnecessary background scores. When $\mathcal{L}_{PA}$ is added without the residual technique, although it contains information about the same part with a different situation or identity, certain specific parts still retain unnecessary information, such as the background or occlusion area. This observation indicates that some parts have learned to carry attention scores for irrelevant information from global attention. When the complete CPA framework is applied, the attention remains consistent across all situations, effectively suppressing extraneous information. This shows that CPA eliminates occlusions and misalignments, providing consistent attention regardless of various contexts.

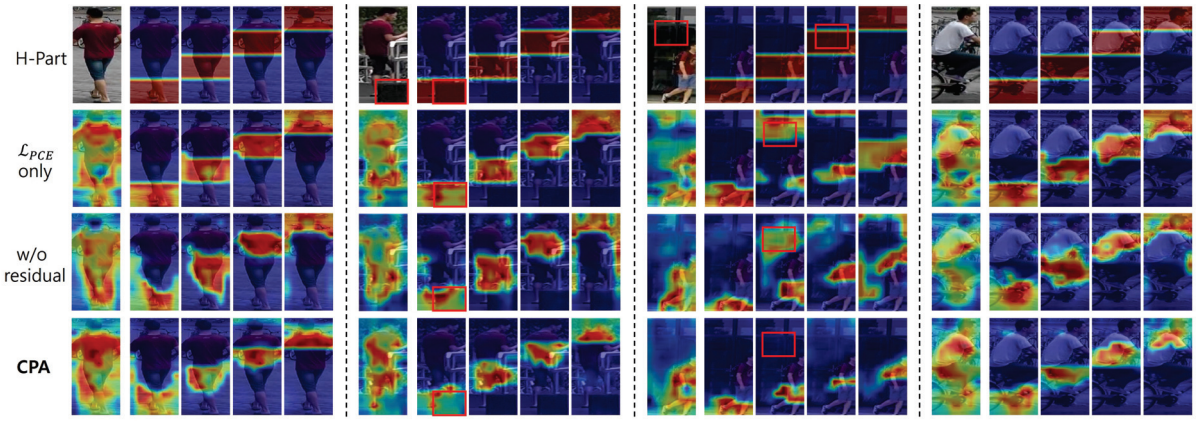


Figure 7. Comparison of attention map visualizations for consistent part-attention (CPA) analysis between different situations (e.g., general, occlusion, misaligned, and pose variations). When applying the full CPA, each situation displays a proper and uniformly divided attention map.

Table 5. Comparison between part-based method and consistent part attention (CPA) in MARS [2]. **Bold** denotes best performance.

Model	mAP	Rank-1	Rank-5	Rank-10
Baseline	85.2	89.1	86.7	97.5
Part 4	85.5	89.9	96.9	97.7
Part 8	85.6	90.1	96.9	97.9
CPA 2	86.5	90.7	97.0	97.8
CPA 4	**86.7**	91.4	97.0	97.9
CPA 8	**86.7**	**91.6**	**97.1**	**98.1**

4.3.2. Comparison of Part-Based and CPA Models

We conducted experiments to compare the proposed CPA framework with the previous part-based approach. With the same MGA module, we only separated the components of different partitioning methods and the number of parts. The results are presented in Table 5. Compared with the baseline, the CPA model with 4 partitions achieved a 1.5%/2.3%

mAP/Rank-1 increment on MARS. Compared with the part-based approach with the same 4 partitions, the model achieved a 1.2%/1.5% mAP/Rank-1 performance increment. The model achieved the best performance for 8 partitions of CPA with a 1.5%/2.5% mAP/Rank-1 increment compared to the baseline. All CPA partitions achieved more remarkable performance than previous horizontal part-based approaches. This result indicates that the proposed CPA method is more informative than the simple horizontal partition by suppressing unnecessary background information and representing only relevant regions. As shown in Figure 8, depending on the number of parts, each attention part uniformly focuses on the most salient semantic part (e.g., Figure 8c, each attention focuses on a semantic part, such as the head, shirt, pants, and shoe). However, if attention parts are separated too much, each part has very little information, making it difficult to determine the semantic area, as depicted in the third image in Figure 8d. The superiority of the proposed CPA approach over the part-based method is observable in the first and last rows in Figure 7.

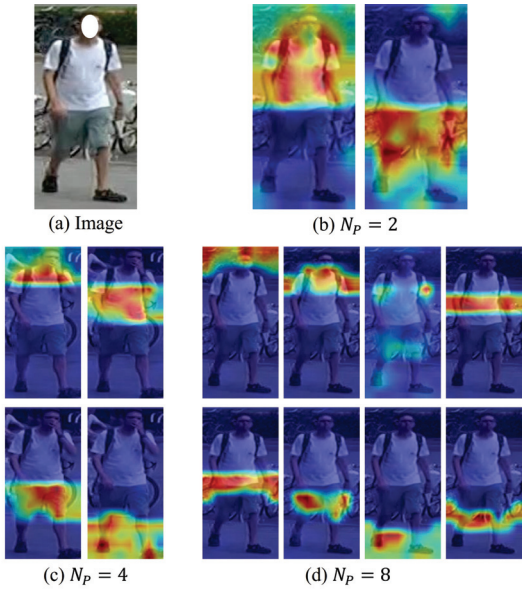


Figure 8. Visualization of part attention from the consistent part-attention (CPA) module with a different number of partitions N_p.

4.3.3. The Influence of Granularity

To assess the effectiveness of MA-PA and LS-TA as a function of granularity, we compared different combinations of granularity scales in the spatial and temporal dimensions. Table 6 shows the details. First, following [15], we experimented with spatially scaling parts by a factor of 2. The Rank-1 performance increases steadily when more detailed spatial granularities are captured. The scaling factors (i.e., 1, 2, 4, 8) resulted in the highest Rank-1 performance with a 2.8% increment compared to the baseline, showing that the MA-PA approach is effective even with deep granularity. This indicates that the model is stronger in feature extraction when semantic information is diversified. Next, we experimented with different granular factors over temporal cues. We observed that the performance saturates when using the optimal granular in the spatial (i.e., 1, 2, 4) and temporal (i.e., 1, 3, 5) domains, respectively. We achieved a performance increment of 2.0%/2.9% mAP/Rank-1. This indicates that the model supplemented it using useful information when various sampling information was delivered over time.

Table 6. Comparison between 'spatial' and 'temporal' scaling factors under different granularity parts in MARS [2]. **Bold** denotes best performance.

Spatial	Temporal	mAP	Rank-1	Rank-5	Rank-10
-	-	85.2	89.1	86.7	97.5
1, 2	1	86.5	90.7	97.0	97.8
1, 2, 4	1	**86.7**	91.1	**97.5**	**98.0**
1, 2, 4, 8	1	**86.7**	**91.9**	97.2	97.8
1, 2, 4	1, 3	86.8	90.6	97.2	97.9
1, 2, 4	1, 3, 5	**87.2**	**92.0**	97.3	98.1
1, 2, 4, 8	1, 3	87.0	90.9	**97.4**	**98.1**
1, 2, 4, 8	1, 3, 5	86.9	91.2	**97.4**	98.0

4.3.4. Effectiveness of ST-MGA

In ST-MGA, we aggregated the partial features in the spatial and temporal domains. First, the partwise aggregator was used to aggregate across part features, and the scalewise aggregator combines across the granularity scales. We first compared different components of the MGA to validate the partwise and scalewise aggregators. As listed in Table 7, we validate different MGA components on the presence or absence of part-specific aggregators ('part') and scale-specific aggregators ('scale'). A 0.9%/0.5% mAP/Rank-1 performance increment occurred when using only the partwise aggregator, and a 0.7%/0.4% mAP/Rank-1 performance increment occurred when using only the scalewise aggregator. Combining both improved the 0.9%/1.5% mAP/Rank-1 performance, indicating that the proposed combination is more effective than simply averaging the partial features (1st row).

Table 7. Performance of ST-MGA in MARS [2] on the presence or absence of partwise aggregator (**'Part'**) and scalewise aggregator (**'Scale'**) in MGA architecture. **Bold** denotes best performance.

Part	Scale	mAP	Rank-1	Rank-5	Rank-10
		86.3	90.5	97.1	97.8
✓		**87.2**	91.0	**97.3**	**98.1**
	✓	87.0	90.9	**97.3**	**98.1**
✓	✓	**87.2**	**92.0**	**97.3**	98.0

To further validate ST-MGA, we conducted comparative experiments on the influence of each MGA in the spatial (S-MGA) and temporal (T-MGA) domains. Table 8 presents the details. Compared with the baseline, S-MGA achieved a 1.7%/1.8% mAP/Rank-1 increment on MARS. Compared to a simple horizontal partition (P-MGA), S-MGA performed better with a 1.0%/0.4% mAP/Rank-1 increment. Combining T-MGA and P-MGA showed little performance difference compared with T-MGA. The reason may be that T-MGA does not work well for simple horizontal parts due to problems such as temporal misalignment. In contrast, when T-MGA and S-MGA were used together, a 1.1%/1.2% mAP/Rank-1 performance increment occurred compared to only using T-MGA. This result is because S-MGA only deals with semantic information through CPA, so the effect of T-MGA is complementary. To verify which order is better between T-MGA and S-MGA, we made comparisons of the above two different orders. For the Rank-1 metric only, T-MGA $\rightarrow$ S-MGA outperformed by 0.5% compared to using S-MGA $\rightarrow$ T-MGA. As a result, it was verified that the proposed model was effective by exploring knowledge in the spatial and temporal domains, regardless of order, and we used T-MGA $\rightarrow$ S-MGA with relatively high Rank-1 performance as the optimal model.

Table 8. Performance of ST-MGA in MARS [2] under different spatial and temporal combinations. **Bold** denotes best performance.

Model	mAP	Rank-1	Rank-5	Rank-10
Baseline	85.2	89.1	86.7	97.5
P-MGA	85.9	90.5	96.5	97.6
S-MGA	86.9	90.9	97.2	**98.1**
T-MGA	86.1	90.8	97.0	97.8
T-MGA → P-MGA	86.0	90.9	96.9	97.8
S-MGA → T-MGA	**87.2**	91.5	**97.3**	**98.1**
T-MGA → S-MGA	**87.2**	**92.0**	**97.3**	98.0

4.4. Comparison and Visualization

4.4.1. Comparison with the State of the Art

As presented in Table 9, we compared ST-MGA with state-of-the-art methods on the MARS, Duke-V, and LS-VID datasets. For the MARS [2] dataset, ST-MGA reached 87.2% mAP and 92.0% Rank-1 results, which is the best mAP and Rank-5 accuracy, and achieved the second-best Rank-1 results. For the Duke-V [68] dataset, the best mAP and Rank-5 accuracy were achieved at 96.8% and 99.9%, respectively. For the LS-VID [27] dataset, the results were competitive, with the third-best mAP and Rank-1 at 79.3% and 88.5%, respectively. We noticed that ST-MGA outperformed ST-GCN [14] and MGH [15], which use a horizontal part-based approach. Specifically, the proposed method also outperformed TCLNet [18] and BiCnet-TKS [42], which usesimilar diverse attention-based methods, with an improvement of up to 1.4%/2.2% and 1.2%/1.8% mAP/Rank-1 accuracy in MARS, respectively. Further, ST-MGA outperformed several recent models (i.e., SINet [37], CAVIT [38], HMN [40], SGMN [41], and BIC+LGCN [42]). In particular, the proposed method shows higher accuracy than the complex transformer-based method [38], which has recently attracted attention. The above results verify the effectiveness and superiority of ST-MGA in video ReID.

Table 9. Quantitative comparison with state-of-the-art methods. **Red** denotes best performance, and **Blue** and **green** denote the second- and third-best performance, respectively.

Method	MARS			Duke-V			LS-VID		
	mAP	Rank1	Rank5	mAP	Rank-1	Rank-5	mAP	Rank-1	Rank-5
M3D [28]	74.1	84.4	-	-	-	-	40.1	57.7	-
STA [7]	80.8	86.3	95.7	94.9	96.2	**99.3**	-	-	-
GLTR [27]	78.5	87.0	95.8	93.7	96.3	**99.3**	44.3	63.1	**77.2**
RGSA [29]	84.0	89.4	-	95.8	97.2	-	-	-	-
FGRA [30]	81.2	87.3	96.0	-	-	-	-	-	-
MG-RAFA [26]	85.8	90.0	86.7	-	-	-	-	-	-
PhD [31]	85.8	90.0	96.7	-	-	-	-	-	-
STGCN [14]	83.7	90.0	86.4	95.7	97.3	**99.3**	-	-	-
MGH [15]	85.8	90.0	96.7	-	-	-	-	-	-
TCLNet [18]	85.8	89.8	-	96.2	96.9	-	-	-	-
AP3D [32]	85.1	90.1	-	95.6	96.3	-	-	-	
AFA [33]	82.9	90.2	96.6	95.4	97.2	**99.4**	-	-	-
SSN3D [34]	86.2	90.1	96.6	96.3	96.8	98.8	-	-	-
BiCnet-TKS [19]	86.0	90.2	-	96.1	96.3	-	75.1	84.6	-
STMN [35]	84.5	90.5	-	95.9	97.0		69.2	82.1	-
STRF [36]	86.1	90.3	-	96.4	**97.4**	-	-	-	-
SINet [37]	86.2	91.0	-	-	-	-	**79.6**	87.4	-
CAVIT [38]	**87.2**	90.8	-	-	-	-	79.2	**89.2**	-
SANet [39]	86.0	**91.2**	**97.1**	**96.7**	**97.7**	**99.9**	-	-	-
HMN [40]	82.6	88.5	96.2	96.1	96.3	-	-	-	-
SGMN [41]	85.4	90.8	-	96.3	96.9	-	-	-	-
BIC+LGCN [42]	**86.5**	91.1	**97.2**	**96.5**	97.1	98.8	-	-	-
IRNet-V [20]	**87.0**	**92.5**	-	-	-	-	**80.5**	**89.4**	-
ST-MGA (ours)	**87.2**	**92.0**	**97.3**	**96.8**	**97.6**	**99.9**	**79.3**	**88.5**	**96.1**

4.4.2. Visualization Analysis

In this section, we visualize some retrieval results from MARS [2] in Figure 9. As can be observed, it is difficult for the baseline model to accurately distinguish people who have a similar appearance or interference when there are an inaccurate detection, occlusion, and uncertain pose. In these cases, the baseline model is not properly focused on the semantic area, and includes unnecessary information about interferences and noises, resulting in relatively low Rank-1 accuracy. The ST-MGA, on the other hand, focused on semantic areas in each part and aggregated them, resulting in high Rank-1 accuracy, even when exposed to relatively difficult and complex interference and noise.

Moreover, we visualize part attention using the proposed CPA module in Figure 10. As observed, the attention parts from CPA focus on similar semantic areas, regardless of the differences between datasets. They each focus on different semantic parts of the face, top, bottom, and shoe and uniformly provide part cues. In occlusion situations, such as in Duke-V, we explored only the correct semantic area, excluding the occluded part. In particular, by focusing on the same semantic part, we can eliminate unnecessary interference in the temporal domain. This allows ST-MGA to contribute to complementary feature extraction by minimizing the interference in both the spatial and temporal domains.

Figure 9. The Visualization of ReID retrieval results using the baseline and the proposed ST-MGA on MARS. For each row, the first sequence is the query, while the five sequences in the middle correspond to the Rank-1 to Rank-5 of the baseline model, and the rest are the retrieval results of our ST-MGA. The correct and incorrect matches are marked with green and red bounding boxes, respectively.

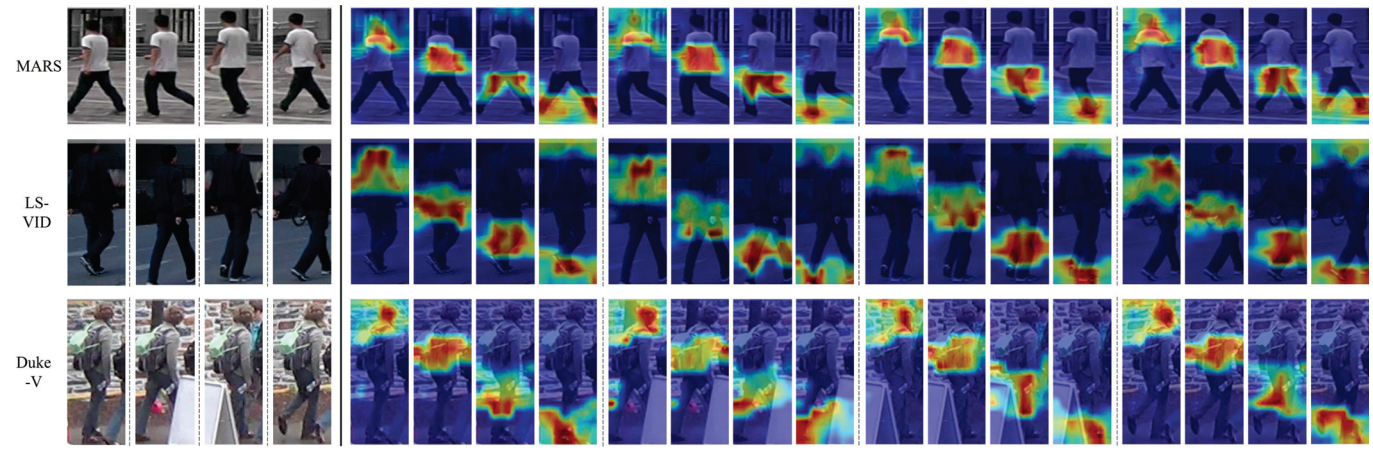


Figure 10. Visualization of part attentions from the consistent part-attention (CPA) module on MARS, LS-VID, and Duke-V. Showing the original image and four separated part attentions for each image for different datasets.

5. Conclusions

This paper presents the ST-MGA framework, a novel approach for robust video-based person ReID. ST-MGA effectively captures consistent spatiotemporal information to mitigate interference and enhances feature extraction through comprehensive utilization of diverse spatiotemporal granular information. To tackle interference arising from spatiotemporal inconsistency, we introduce the CPA module, which learns from self-information and specific priors. The CPA module efficiently separates attention parts to extract features with spatially uniform amounts and temporally identical semantic information. Additionally, our approach employs multi-granularity feature augmentation to synthesize granular information encompassing semantic attention parts across various scales. Spatially, MA-PA extracts various semantic granular information by synthesizing fine attention without overlapping. Temporally, LS-TA augmented various granular features through various time sampling intervals. Leveraging granular information with different scales, the MGA module effectively utilizes spatiotemporal cues to extract complementary features. Within MGA, we explored the relationships between part/scale information, aggregating them based on relation scores. The resulting aggregated representations enable complementary feature extraction by prioritizing pertinent semantic information while filtering out unnecessary interference or noise. Extensive experiments corroborated the effectiveness and superiority of our ST-MGA, highlighting its potential for advancing video-based person ReID research.

Author Contributions: Conceptualization, H.S.L.; Methodology, H.S.L.; Software, H.S.L.; Validation, H.S.L. and M.K.; Formal analysis, M.K. and S.J.; Writing—original draft, H.S.L.; Writing—review & editing, M.K., S.J. and H.B.B.; Visualization, H.B.B.; Supervision, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0 02068, Artificial Intelligence Innovation Hub).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, K.; Ma, B.; Zhang, W.; Huang, R. A spatio-temporal appearance representation for viceo-based pedestrian re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3810–3818.
2. Zheng, L.; Bie, Z.; Sun, Y.; Wang, J.; Su, C.; Wang, S.; Tian, Q. Mars: A video benchmark for large-scale person re-identification. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VI 14; Springer: Cham, Switzerland, 2016; pp. 868–884.
3. Liu, Z.; Wang, Y.; Li, A. Hierarchical integration of rich features for video-based person re-identification. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 3646–3659. [CrossRef]
4. Kim, M.; Cho, M.; Lee, S. Feature Disentanglement Learning with Switching and Aggregation for Video-based Person Re-Identification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 1603–1612.
5. Chen, D.; Li, H.; Xiao, T.; Yi, S.; Wang, X. Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1169–1178.
6. McLaughlin, N.; Del Rincon, J.M.; Miller, P. Recurrent convolutional network for video-based person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1325–1334.
7. Fu, Y.; Wang, X.; Wei, Y.; Huang, T. Sta: Spatial-temporal attention for large-scale video-based person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8287–8294.

8. Xu, S.; Cheng, Y.; Gu, K.; Yang, Y.; Chang, S.; Zhou, P. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4733–4742.
9. Liu, Y.; Yan, J.; Ouyang, W. Quality aware network for set to set recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5790–5799.
10. Gao, J.; Nevatia, R. Revisiting temporal modeling for video-based person reid. *arXiv* **2018**, arXiv:1805.02104.
11. Li, S.; Bak, S.; Carr, P.; Wang, X. Diversity regularized spatiotemporal attention for video-based person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 369–378.
12. Zhao, Y.; Shen, X.; Jin, Z.; Lu, H.; Hua, X.S. Attribute-driven feature disentangling and temporal aggregation for video person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4913–4922.
13. Liu, Y.; Yuan, Z.; Zhou, W.; Li, H. Spatial and temporal mutual promotion for video-based person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8786–8793.
14. Yang, J.; Zheng, W.S.; Yang, Q.; Chen, Y.C.; Tian, Q. Spatial-temporal graph convolutional network for video-based person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3289–3299.
15. Yan, Y.; Qin, J.; Chen, J.; Liu, L.; Zhu, F.; Tai, Y.; Shao, L. Learning multi-granular hypergraphs for video-based person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2899–2908.
16. Wu, Y.; Bourahla, O.E.F.; Li, X.; Wu, F.; Tian, Q.; Zhou, X. Adaptive graph representation learning for video person re-identification. *IEEE Trans. Image Process.* **2020**, *29*, 8821–8830. [CrossRef] [PubMed]
17. Liu, C.T.; Wu, C.W.; Wang, Y.C.F.; Chien, S.Y. Spatially and temporally efficient non-local attention network for video-based person re-identification. *arXiv* **2019**, arXiv:1908.01683.
18. Hou, R.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Temporal complementary learning for video person re-identification. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXV 16; Springer: Cham, Switzerland, 2020; pp. 388–405.
19. Hou, R.; Chang, H.; Ma, B.; Huang, R.; Shan, S. Bicnet-tks: Learning efficient spatial-temporal representation for video person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2014–2023.
20. Tao, H.; Duan, Q.; An, J. An Adaptive Interference Removal Framework for Video Person Re-Identification. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 5148–5159. [CrossRef]
21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
22. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
23. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
24. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
25. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
26. Zhang, Z.; Lan, C.; Zeng, W.; Chen, Z. Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10407–10416.
27. Li, J.; Wang, J.; Tian, Q.; Gao, W.; Zhang, S. Global-local temporal representations for video person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3958–3967.
28. Li, J.; Zhang, S.; Huang, T. Multi-scale 3d convolution network for video based person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8618–8625.
29. Li, X.; Zhou, W.; Zhou, Y.; Li, H. Relation-guided spatial attention and temporal refinement for video-based person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11434–11441.
30. Chen, Z.; Zhou, Z.; Huang, J.; Zhang, P.; Li, B. Frame-guided region-aligned representation for video person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10591–10598.
31. Zhao, J.; Qi, F.; Ren, G.; Xu, L. Phd learning: Learning with pompeiu-hausdorff distances for video-based vehicle re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2225–2235.

32. Gu, X.; Chang, H.; Ma, B.; Zhang, H.; Chen, X. Appearance-preserving 3d convolution for video-based person re-identification. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16; Springer: Cham, Switzerland, 2020; pp. 228–243.
33. Chen, G.; Rao, Y.; Lu, J.; Zhou, J. Temporal coherence or temporal motion: Which is more critical for video-based person re-identification? In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part VIII 16; Springer: Cham, Switzerland, 2020; pp. 660–676.
34. Jiang, X.; Qiao, Y.; Yan, J.; Li, Q.; Zheng, W.; Chen, D. SSN3D: Self-separated network to align parts for 3D convolution in video person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35; pp. 1691–1699.
35. Eom, C.; Lee, G.; Lee, J.; Ham, B. Video-based person re-identification with spatial and temporal memory networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12036–12045.
36. Aich, A.; Zheng, M.; Karanam, S.; Chen, T.; Roy-Chowdhury, A.K.; Wu, Z. Spatio-temporal representation factorization for video-based person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 152–162.
37. Bai, S.; Ma, B.; Chang, H.; Huang, R.; Chen, X. Salient-to-broad transition for video person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7339–7348.
38. Wu, J.; He, L.; Liu, W.; Yang, Y.; Lei, Z.; Mei, T.; Li, S.Z. CAViT: Contextual Alignment Vision Transformer for Video Object Re-identification. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 549–566.
39. Bai, S.; Ma, B.; Chang, H.; Huang, R.; Shan, S.; Chen, X. SANet: Statistic attention network for video-based person re-identification. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 3866–3879. [CrossRef]
40. Wang, Z.; He, L.; Tu, X.; Zhao, J.; Gao, X.; Shen, S.; Feng, J. Robust video-based person re-identification by hierarchical mining. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 8179–8191. [CrossRef]
41. Chen, C.; Ye, M.; Qi, M.; Wu, J.; Liu, Y.; Jiang, J. Saliency and granularity: Discovering temporal coherence for video-based person re-identification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6100–6112. [CrossRef]
42. Pan, H.; Liu, Q.; Chen, Y.; He, Y.; Zheng, Y.; Zheng, F.; He, Z. Pose-Aided Video-based Person Re-Identification via Recurrent Graph Convolutional Network. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 7183–7196. [CrossRef]
43. He, T.; Jin, X.; Shen, X.; Huang, J.; Chen, Z.; Hua, X.S. Dense interaction learning for video-based person re-identification supplementary materials. *Identities* **2021**, *1*, 300.
44. Zhou, Z.; Huang, Y.; Wang, W.; Wang, L.; Tan, T. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4747–4756.
45. Liao, X.; He, L.; Yang, Z.; Zhang, C. Video-based person re-identification via 3d convolutional networks and non-local attention. In Proceedings of the Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Revised Selected Papers, Part VI 14; Springer: Cham, Switzerland, 2019; pp. 620–634.
46. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef] [PubMed]
47. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
48. Song, G.; Leng, B.; Liu, Y.; Hetang, C.; Cai, S. Region-based quality estimation network for large-scale person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
49. Chen, B.; Deng, W.; Hu, J. Mixed high-order attention network for person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 371–381.
50. Wang, C.; Zhang, Q.; Huang, C.; Liu, W.; Wang, X. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 365–381.
51. Zhao, L.; Li, X.; Zhuang, Y.; Wang, J. Deeply-learned part-aligned representations for person re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3219–3228.
52. Li, W.; Zhu, X.; Gong, S. Harmonious attention network for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2285–2294.
53. Chen, H.; Zhao, Y.; Wang, S. Person Re-Identification Based on Contour Information Embedding. *Sensors* **2023**, *23*, 774. [CrossRef] [PubMed]
54. Zhang, Z.; Lan, C.; Zeng, W.; Jin, X.; Chen, Z. Relation-aware global attention for person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3186–3195.
55. Chen, T.; Ding, S.; Xie, J.; Yuan, Y.; Chen, W.; Yang, Y.; Ren, Z.; Wang, Z. Abd-net: Attentive but diverse person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8351–8361.

56. Kim, M.; Cho, M.; Lee, H.; Cho, S.; Lee, S. Occluded person re-identification via relational adaptive feature correction learning. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 2719–2723.
57. Xu, J.; Zhao, R.; Zhu, F.; Wang, H.; Ouyang, W. Attention-aware compositional network for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2119–2128.
58. Song, C.; Huang, Y.; Ouyang, W.; Wang, L. Mask-guided contrastive attention model for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1179–1188.
59. Si, J.; Zhang, H.; Li, C.G.; Kuen, J.; Kong, X.; Kot, A.C.; Wang, G. Dual attention matching network for context-aware feature sequence based person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5363–5372.
60. Liu, J.; Zha, Z.J.; Wu, W.; Zheng, K.; Sun, Q. Spatial-temporal correlation and topology learning for person re-identification in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4370–4379.
61. Tao, H.; Lu, M.; Hu, Z.; Xin, Z.; Wang, J. Attention-aggregated attribute-aware network with redundancy reduction convolution for video-based industrial smoke emission recognition. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7653–7664. [CrossRef]
62. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
63. Wang, Y.; Zhang, P.; Gao, S.; Geng, X.; Lu, H.; Wang, D. Pyramid spatial-temporal aggregation for video-based person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12026–12035.
64. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
65. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
66. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
67. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5533–5541.
68. Wu, Y.; Lin, Y.; Dong, X.; Yan, Y.; Ouyang, W.; Yang, Y. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5177–5186.
69. Zhang, Z.; Lan, C.; Zeng, W.; Chen, Z. Densely semantically aligned person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 667–676.
70. Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; Wang, S. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 480–496.
71. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York NY, USA, 7–12 February 2020; Volume 34, pp. 13001–13008.
72. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
73. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 20 February 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Infrared Image Super-Resolution Network Utilizing the Enhanced Transformer and U-Net

Feng Huang, Yunxiang Li, Xiaojing Ye and Jing Wu *

School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350108, China; huangf@fzu.edu.cn (F.H.); 17864285690@163.com (Y.L.); yexiaojingyxj@163.com (X.Y.)
* Correspondence: wujing@fzu.edu.cn

Abstract: Infrared images hold significant value in applications such as remote sensing and fire safety. However, infrared detectors often face the problem of high hardware costs, which limits their widespread use. Advancements in deep learning have spurred innovative approaches to image super-resolution (SR), but comparatively few efforts have been dedicated to the exploration of infrared images. To address this, we design the Residual Swin Transformer and Average Pooling Block (RSTAB) and propose the SwinAIR, which can effectively extract and fuse the diverse frequency features in infrared images and achieve superior SR reconstruction performance. By further integrating SwinAIR with U-Net, we propose the SwinAIR-GAN for real infrared image SR reconstruction. SwinAIR-GAN extends the degradation space to better simulate the degradation process of real infrared images. Additionally, it incorporates spectral normalization, dropout, and artifact discrimination loss to reduce the potential image artifacts. Qualitative and quantitative evaluations on various datasets confirm the effectiveness of our proposed method in reconstructing realistic textures and details of infrared images.

Keywords: transformer; generative adversarial network; infrared image; image super-resolution

1. Introduction

Infrared imaging utilizes infrared detectors to capture images, differentiating target objects from the background based on the disparities found in radiated heat. This technique effectively addresses the constraints encountered in visible imaging, such as limited object penetration and varying weather conditions, thus playing a pivotal role in diverse fields including climate research [1], disaster warning [2], rescue [3] and resource exploration [4]. However, the resolution of infrared images depends on the quantity and size of the dedicated infrared detector array. The hardware limitation can restrict the imaging quality.

Image super-resolution (SR) can transform low-resolution (LR) images into high-resolution (HR) ones without enhancing optical detector sensitivity or developing complex optical imaging systems, significantly reducing costs. Recent advancements in deep learning have demonstrated exceptional performance in utilizing SR for image enhancement.

Early deep learning-based image SR methods (e.g., SRCNN [5], VDSR [6], and EDSR [7]) rely on convolutional neural network (CNN) to learn the mapping from LR images to HR images. However, since convolutional kernels can only perceive local information from the input image, it is necessary to stack deep networks to gradually enlarge the receptive field. Therefore, CNN may be limited in handling global information in images. Recently, transformer-based methods [8–12] have become widely popular in the field of image SR. They leverage a self-attention mechanism to effectively capture long-range dependencies and handle multi-scale features in images, thus achieving better SR performance. Despite this, the above methods rely only on custom degradation models, such as bicubic interpolation (BI), for paired training, which tend to overlook the complex degradation processes present in real-world images, resulting in overly smooth SR results.

Nowadays, blind-image SR networks (e.g., BSRGAN [13] and Real-ESRGAN [14]) have gained attention for their ability to handle real-world SR tasks. These methods generate training pairs using unknown degradations, which leads to more robust and generalizable models that better meet real-world requirements. However, these works primarily focus on visible images. Few studies have extended image SR to the field of infrared images.

To address the SR challenge in infrared images, we design the Residual Swin Transformer and Average Pooling Block (RSTAB) and construct the novel Infrared Image Super-Resolution model based on Swin Transformer [8] and Average Pooling [15], SwinAIR. SwinAIR combines the strengths of CNN and transformer, allowing for the sufficient extraction of features from infrared images. To achieve SR reconstruction of real infrared images, we further integrate the SwinAIR with the generative adversarial network (GAN) and build the SwinAIR-GAN, which can recover realistic textures and features matching the real infrared images. The primary contributions of this study are as follows:

- We explore a novel infrared image SR reconstruction model, SwinAIR. We introduce the Residual Swin Transformer and Average Pooling Block into the deep feature extraction module of SwinAIR. This configuration effectively extracts both low- and high-frequency infrared features while concurrently fusing them. Comparative results with existing methods demonstrate that our model exhibits superior performance in the SR reconstruction of infrared images.
- We combine SwinAIR with the U-Net [16] to construct SwinAIR-GAN, further exploring the problem of SR reconstruction for real infrared images. SwinAIR-GAN utilizes SwinAIR as the generator network and employs U-Net as the discriminator network. Comparative results with similar methods demonstrate that our model can generate infrared images with better visual effects and more realistic features and details.
- We incorporate spectral normalization, dropout, and artifact discrimination loss to minimize possible artifacts during the restoration process of real infrared images and enhance the generalization ability of SwinAIR-GAN. We also expand the degradation space of the degradation model to emulate the degradation process of real infrared images more accurately. These improvements enable the generation of infrared images that closely align with the textures and details of real-world images while avoiding over-smoothing.
- We establish an infrared data acquisition system that can simultaneously capture LR and HR infrared images of corresponding scenes, addressing the issue of lacking reference images in the real infrared image SR reconstruction task.

The rest of this paper is structured as follows. Section 2 provides a brief overview of related studies, and Section 3 offers a detailed description of the proposed network. In Section 4, we validate the effectiveness of our network by testing it on various datasets and conducting ablation experiments. Finally, the conclusions are presented in Section 5.

2. Related Works

Infrared image SR methods are derived from visible image SR methods. In this section, we briefly review the image SR reconstruction research status related to our work.

2.1. Traditional Infrared Image Super-Resolution Reconstruction Methods

Traditional infrared image SR methods, similar to visible image SR methods, can be categorized into frequency domain-based, dictionary-based, and other methods. Frequency domain-based methods decompose the infrared image into the spatial domain and the frequency domain [17], processing the image separately in each domain. For example, Choi et al. [18] distinguished the edge pixels of infrared images and utilized visible images as guidance to enhance the high-frequency information of infrared images. Mao et al. [19] transformed the infrared image SR problem into a sparse signal reconstruction problem based on differential operations, significantly enhancing the algorithm's speed. Dictionary-based methods [20–25] focus on constructing the mapping relationship between LR and

HR image patches. By leveraging the relationships between image patches, these methods retain the pattern information of LR images in HR images. This type of method has garnered researchers' attention for its interpretability. Other traditional methods include iterative methods [26,27], regularization methods [28], and so on.

With the advancement of deep learning technology, transformer and GAN technologies have begun to be widely applied in the field of image SR reconstruction, demonstrating significant advantages.

2.2. Transformer-Based Image Super-Resolution Reconstruction Methods

Transformer was initially proposed by Vaswani et al. [29] to address sequence modeling in natural language processing tasks. The self-attention mechanism allows the transformer to effectively capture global dependencies. Additionally, its parallel computing capability accelerates training and enhances efficiency. With the development of vision [30,31], detection [32,33], and video-vision [34,35] transformers, superior results have been achieved by transformers in various visual tasks.

In the field of image SR reconstruction, there are mainly two types of transformer-based frameworks. The first type utilizes the transformer structure entirely, such as the end-to-end network proposed by Chen et al. [9]. However, this framework may lack adaptability to biases in some cases. The second type combines transformer with CNN, leveraging the strengths of CNN in capturing local features and transformer in modeling global dependency relationships. This integration enables better capture of both local details and global structures in the image. For example, Liu et al. [8] introduced a backbone Swin Transformer that replaces the standard multi-head self-attention with a shifted-window multi-head self-attention for dense hierarchical feature map prediction, enabling efficient interactions among non-overlapping local windows while addressing the challenges posed by larger-scale images and huge training datasets. Liang et al. [10] built upon this idea and proposed SwinIR, which has demonstrated impressive performance in various tasks, including image SR and dehazing. Subsequently, researchers have continuously produced new research results [11,12] with better performance based on this foundation. Cao et al. [36] took SwinIR as the architecture and proposed the CFMB-T model for infrared remote sensing images SR reconstruction. Yi et al. [37] further utilized a hybrid convolution–transformer network for feature extraction, and the suggested HCTIR deblur model has made great progress in infrared image deblurring.

Despite the promising performance of transformer-based methods in image SR reconstruction, strategies that combine the transformer with CNN for deep feature extraction are still limited. This limitation often results in networks overlooking high-frequency information present in images. The problem is particularly pronounced when dealing with infrared images that involve blurring and a lack of high-frequency details. To address this issue, we propose SwinAIR, which better integrates the transformer with CNN to effectively capture both low- and high-frequency information, thereby enhancing the SR reconstruction capability for infrared images.

2.3. GAN-Based Image Super-Resolution Reconstruction Methods

GAN was proposed by Goodfellow et al. in 2014 [38]. It consists of a generator network and a discriminator network that are trained adversarially. The generator network progressively creates more realistic samples, while the discriminator network continuously refines its ability to differentiate between real and artificial images. Training concludes when the discriminator network struggles at a given threshold to distinguish between the two, resulting in high-quality generated samples.

Ledig et al. [39] first applied the GAN to image SR reconstruction and introduced the SRGAN. The SRGAN incorporates perceptual loss, which provides more detailed and lifelike SR images. Subsequently, Yan et al. [40] optimized the image quality assessment approach and used it to improve the loss function of SRGAN, further enhancing its performance. Wang et al. [41] introduced the ESRGAN model in 2018, which incorporates

a residual dense block into its generator network and employs a relative discriminator network. By enhancing perceptual loss and introducing interpolation, ESRGAN achieves exceptional perceptual reconstruction effects. Based on the ESRGAN, several newer networks have been developed, including BSRGAN [13], RFB-ESRGAN [42], and Real-ESRGAN [14].

Liu et al. [43] introduced the natural image gradient prior. They used visible images from corresponding scenes as guidance to generate infrared images with improved subjective visual quality. Huang et al. [44] proposed HetSRWGAN, which incorporates heterogeneous convolution and adversarial training. The reconstructed infrared images using this method achieve higher objective evaluation metrics. They also introduced the lightweight PSGAN [45] using a multi-stage transfer learning strategy. Liu et al. [46] designed a generator network with recursive attention modules, significantly improving the SR quality of vehicle infrared images. Lee et al. [47] proposed the Style Transfer Super-Resolution Generative Adversarial Network, STSRGAN, enhancing the quality of LR infrared images containing very small targets.

Although the aforementioned GAN-based methods have shown high potentiality, their training processes are often unstable and result in unintended artifacts. To generate more realistic infrared images, we employ the spectral normalization-based U-Net [16] as the discriminator network and add the artifact discrimination loss function in our SwinAIR-GAN.

3. Proposed Method

In this section, we will comprehensively introduce the SwinAIR and SwinAIR-GAN along with their technical details.

3.1. SwinAIR

3.1.1. Overall Structure

The structure of SwinAIR is displayed in Figure 1. Infrared images generally have lower resolution and contain more low-frequency information than visible images. Therefore, the input LR infrared image first undergoes shallow feature extraction and is passed directly to the end of the deep feature extraction network. This helps to better preserve its low-frequency information, thereby improving the performance of SwinAIR.

Concurrently, the shallow features are input to the deep feature extraction module to obtain the deep features. We adopt the hierarchical design of the deep feature extraction module from the Swin Transformer [8]. This structure allows for the extraction and integration of features at different scales in the infrared image, helping the network to better learn critical information and reduce noise.

Ultimately, the shallow features and deep features are merged and fed into the upsampling module to generate the SR infrared image.

In the shallow feature extraction module, the LR infrared image is processed by a 3 × 3 convolutional layer to extract shallow feature information as follows:

$$F_{\mathrm{SF}} = H_{\mathrm{SF}}(I_{\mathrm{LR}}) \tag{1}$$

where I_{LR} denotes the LR infrared image, and $I_{\mathrm{LR}} \in R^{H_L \times W_L \times C_L}$, H_L, W_L, and C_L indicate the height, width, and channel number of the LR infrared image, respectively. $H_{\mathrm{SF}}(\bullet)$ represents the shallow feature extraction module, and F_{SF} represents the shallow feature map.

In the deep feature extraction module, multiple stacked RSTABs (as shown in Figure 1b) and a 3 × 3 convolutional layer are employed to further extract deep features. Residual connections are also used for feature aggregation to obtain deep features, and this process can be simplified as

$$F_{\mathrm{DF}} = H_{\mathrm{DF}}(F_{\mathrm{SF}}) \tag{2}$$

where $H_{\mathrm{DF}}(\bullet)$ corresponds to the deep feature extraction module, and F_{DF} represents the deep feature map. The deep feature extraction module not only expands the receptive field by stacking convolutional layers but also combines the advantages of both transformer

and CNN to effectively extract low- and high-frequency image information. With local and global residual learning, the problem of gradient degradation in deep networks is mitigated, significantly reducing the parameters and overall training time.

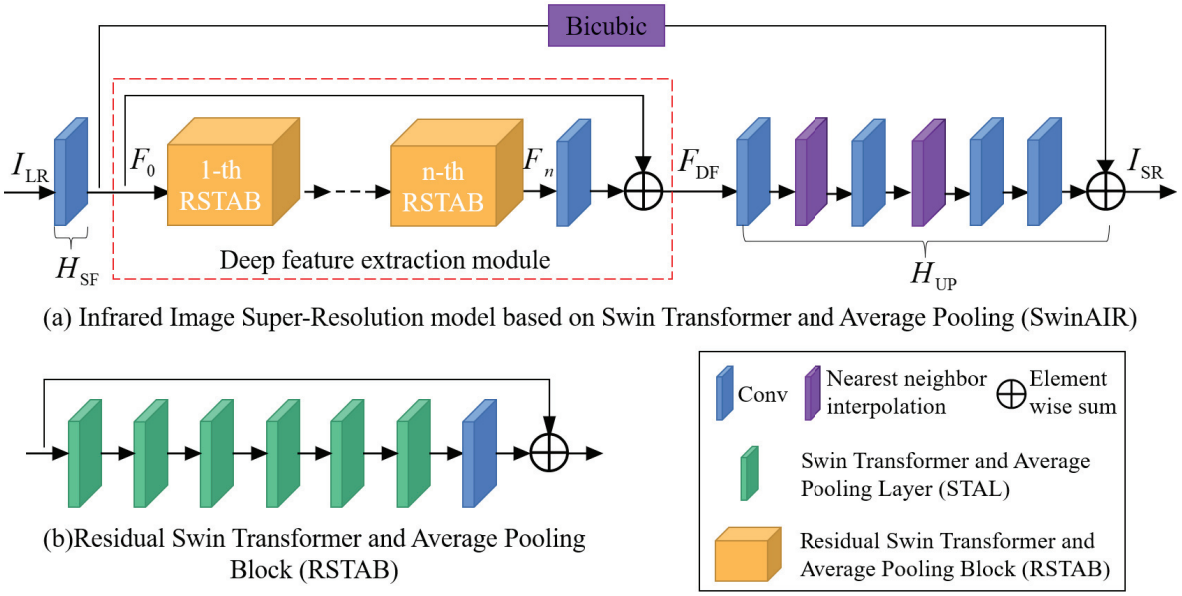


Figure 1. Structure of our proposed Infrared Image Super-Resolution model based on Swin Transformer and Average Pooling, SwinAIR. Firstly, the input LR infrared I_{LR} undergoes the shallow feature extraction module H_{SF} to extract the shallow feature map F_0; then, F_0 is deep extracted and refined by the deep feature extraction module. We obtain F_n after n RSTABs. After F_n undergoes convolutional operations, it is further combined with F_0 through residual connection to obtain the deep feature map F_{DF}. Finally, F_{DF} is input into the upsampling module H_{UP} to generate the output SR infrared image I_{SR}. The interpolation operation here is bicubic.

Finally, the feature map obtained from the deep feature extraction module is reconstructed layer by layer according to a specified upsampling factor that follows the order of the upsampling module (as shown in Figure 1a). Simultaneously, the upsampled LR infrared image is moved to the end of the upsampling module, where it is added to the reconstructed feature map to produce the final SR infrared image:

$$I_{SR} = H_{UP}(F_{DF}) \tag{3}$$

where $H_{UP}(\bullet)$ represents the upsampling module, I_{SR} is the SR infrared image, and $I_{SR} \in R^{H_S \times W_S \times C_S}$, H_S, W_S, and C_S indicate the height, width, and channel number of the SR infrared image, respectively (similar to I_{LR}). The specific steps of the upsampling reconstruction process can be expressed as

$$F_{UP} = H_C(D(H_C(H_{Near}(H_C(H_{Near}(H_C(F_{DF})))))))) \tag{4}$$

$$I_{SR} = F_{UP} + H_{Bicubic}(F_{SF}) \tag{5}$$

where $H_C(\bullet)$ denotes a 3 × 3 convolutional layer, $H_{Near}(\bullet)$ represents the nearest neighbor interpolation, $H_{Bicubic}(\bullet)$ denotes the bicubic interpolation, F_{UP} represents the reconstructed feature map and $D(\bullet)$ signifies the dropout operation. Dropout is a regularization technique originally proposed to address overfitting issues in classification networks. A recent study has demonstrated that dropout can enhance the generalizability of SR networks [48]. Therefore, we apply dropout before the last convolutional layer in the upsampling module in SwinAIR with a probability of 0.5. The effectiveness of this approach is confirmed by ablation experiments in Section 4.4.2.

3.1.2. Deep Feature Extraction Module

Drawing inspiration from previous work [8,49], we develop the deep feature extraction module for SwinAIR. It is mainly composed of several RSTABs, which combine the strengths of transformer and CNN to effectively extract low- and high-frequency information from infrared images.

Specifically, the deep feature extraction module consists of n RSTABs and a 3 × 3 convolutional layer with residual connections. The intermediate feature map obtained from the ith RSTAB, F_i, and the final output deep feature map, F_{DF}, are given as follows:

$$F_i = H_{\mathrm{RSTAB}_i}(F_{i-1}), i = 1, 2, \ldots, n \tag{6}$$

$$F_{\mathrm{DF}} = H_{\mathrm{C}}(F_n) + F_0 \tag{7}$$

where F_0 denotes the shallow feature map F_{SF}, and $H_{\mathrm{RSTAB}_i}(\bullet)$ represents the ith RSTAB. By introducing a convolutional layer at the end of the deep feature extraction module, the inductive bias of convolutional operations is incorporated into the transformer, providing a more robust foundation for the subsequent aggregation of shallow and deep features [10].

As illustrated in Figure 1b, each RSTAB is composed of six Swin Transformers and Average Pooling Layers (STALs) and a 3 × 3 convolutional layer with the residual connection. This allows local features to be transmitted and fused across different layers, enabling the network to deeply learn the mapping relationship between LR and HR infrared images, thereby accelerating the network's learning and convergence speed.

For the given input feature map of the ith RSTAB, $F_{i,0}$, the intermediate output feature map can be further defined as follows:

$$F_{i,j} = H_{\mathrm{STAL}_{i,j}}\left(F_{i,j-1}\right), j = 1, 2, \ldots, m \tag{8}$$

$$F_i = H_{\mathrm{C}_i}(F_{i,m}) + F_{i,0} \tag{9}$$

where $H_{\mathrm{STAL}_{i,j}}(\bullet)$ denotes the jth STAL in the ith RSTAB, and $F_{i,j}$ represents the feature map output in the jth STAL of the ith RSTAB. $H_{\mathrm{C}_i}(\bullet)$ represents the final 3 × 3 convolutional layer in the ith RSTAB.

The internal structure of STAL is shown in Figure 2. Each STAL has two sections: the High-Frequency Feature Extraction Layer (HFEL), which consists of two parallel average pooling and fully connected layers, and the Swin Transformer Layer (STL) [8]. The use of channel separation strategies allows the network to independently process low- and high-frequency information in infrared images, mitigating interference during the learning of different structural features. This approach also reduces computational complexity and accelerates the convergence of the network.

Specifically, for the jth STAL of the ith RSTAB, we assume the input feature map $F_{i,j-1} \in R^{H_{i,j-1} \times W_{i,j-1} \times C_{i,j-1}}$. $H_{i,j-1}$, $W_{i,j-1}$, and $C_{i,j-1}$ indicate the height, width, and channel number of $F_{i,j-1}$. $F_{i,j-1}$ is partitioned channel-wise to $F_h \in R^{H_h \times W_h \times C_h}$ and $F_l \in R^{H_l \times W_l \times C_l}$ ($C_h = C_{i,j-1}/3, C_l = 2C_{i,j-1}/3$); then, F_h and F_l are independently fed to the HFEL and the STL.

With the sensitivity of the average pooling layers and the ability of the fully connected layers to capture details [49], the HFEL comprises two parallel paths to perform high-frequency information extraction, each containing an average pooling layer and a fully connected layer. F_h is divided into two channel-wise maps $F_{h1} \in R^{H_h \times W_h \times (C_h/2)}$ and $F_{h2} \in R^{H_h \times W_h \times (C_h/2)}$. F_{h1} and F_{h2} are separately processed in the parallel paths to generate their respective outputs $\hat{F}_{h1}$, $\hat{F}_{h2}$ as follows:

$$\hat{F}_{h1} = FC(Avgpool(F_{h1})) \tag{10}$$

$$\hat{F}_{h2} = FC(Avgpool(F_{h2})) \tag{11}$$

where $Avgpool(\bullet)$ signifies the average pooling layer and $FC(\bullet)$ symbolizes the fully connected layer.

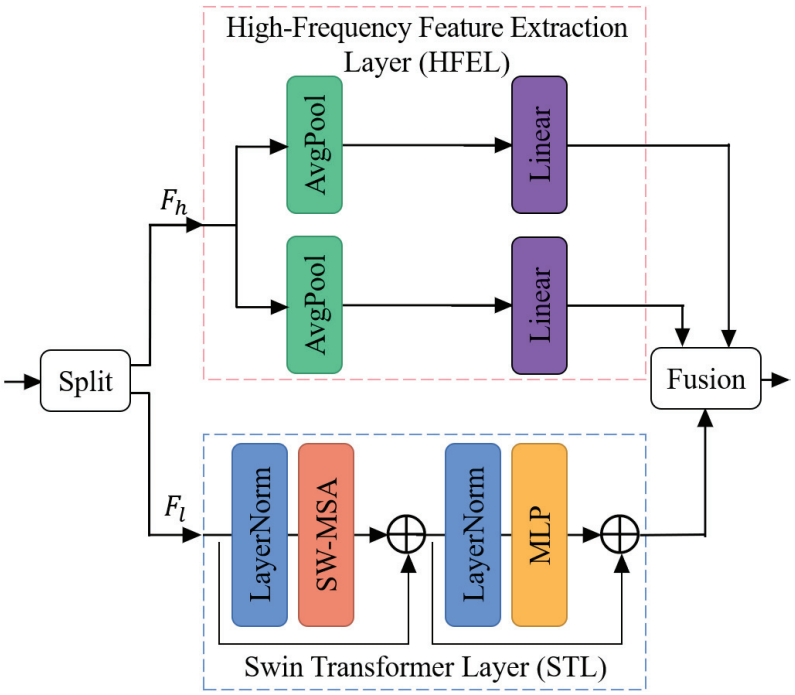


Figure 2. Structure of the Swin Transformer and Average Pooling Layer (STAL). There are two ways to combine CNN with the transformer: serial or parallel. The serial method means that each layer can process either low- or high-frequency information but not both. Therefore, to allow each layer to process both types of information simultaneously, a parallel structure with channel separation is used to integrate the CNN and transformer. The feature map is initially divided into F_h and F_l, which are then separately fed into the High-Frequency Feature Extraction Layer (HFEL) and the Swin Transformer Layer (STL). In STL, SW-MSA and MLP represent the self-attention module based on the shift-window mechanism and the multi-layer perceptron module, respectively.

To decrease the computational burden associated with the standard transformer for image processing, we use STL for low-frequency feature extraction. As shown in Figure 2, STL primarily consists of the self-attention module based on the shift-window mechanism (SW-MSA) and the multi-layer perceptron module (MLP). SW-MSA calculates multi-head self-attention within local regions of the image by shift window. This process captures the correlations between different local regions of the image, thereby extracting local structures and texture information. MLP combines non-linear transformations to better capture the semantic information of the image. For F_l, the processing process can be described as shown below:

$$\hat{F}_l = H_{\mathrm{MLP}}(LN(H_{\mathrm{MSA}}(LN(F_l)) + F_l)) + H_{\mathrm{MSA}}(LN(F_l)) + F_l \tag{12}$$

where $\hat{F}_l$ represents the feature map after processing by the STL. $LN(\bullet)$ denotes the layer normalization operation, H_{MSA} represents the SW-MSA module, and H_{MLP} represents the MLP module.

Finally, the low-frequency information feature map after processing by the STL, $\hat{F}_l$, and the high-frequency information feature maps after processing by the HFEL, $\hat{F}_{h1}$, $\hat{F}_{h2}$, are fused along the channel dimension to get $F_{i,j}$, which can be represented as

$$F_{i,j} = FU(\hat{F}_{h1}, \hat{F}_{h2}, \hat{F}_l) \tag{13}$$

where $FU(\bullet)$ represents the fusion operation.

3.1.3. Loss Function

SwinAIR uses L1 loss as the loss function.The L1 loss calculates the pixel-wise differences between SR and HR images as follows:

$$L_1 = \frac{1}{hwc} \sum_{i,j,k} \left| I^{\mathrm{SR}}_{i,j,k} - I^{\mathrm{HR}}_{i,j,k} \right| \tag{14}$$

where h, w, and c denote the height, width, and channel number of the image, respectively. $I^{\mathrm{SR}}_{i,j,k}$ and $I^{\mathrm{HR}}_{i,j,k}$ represent the pixel values of the SR and HR images at the spatial location, (i, j, k), respectively.

3.2. SwinAIR-GAN

To generate infrared images that more closely align with real physical characteristics, we further design SwinAIR-GAN based on SwinAIR. The proposed SwinAIR-GAN structure for real infrared image SR is depicted in Figure 3. We use SwinAIR as the generator network and a spectral normalization-based U-Net as the discriminator network.

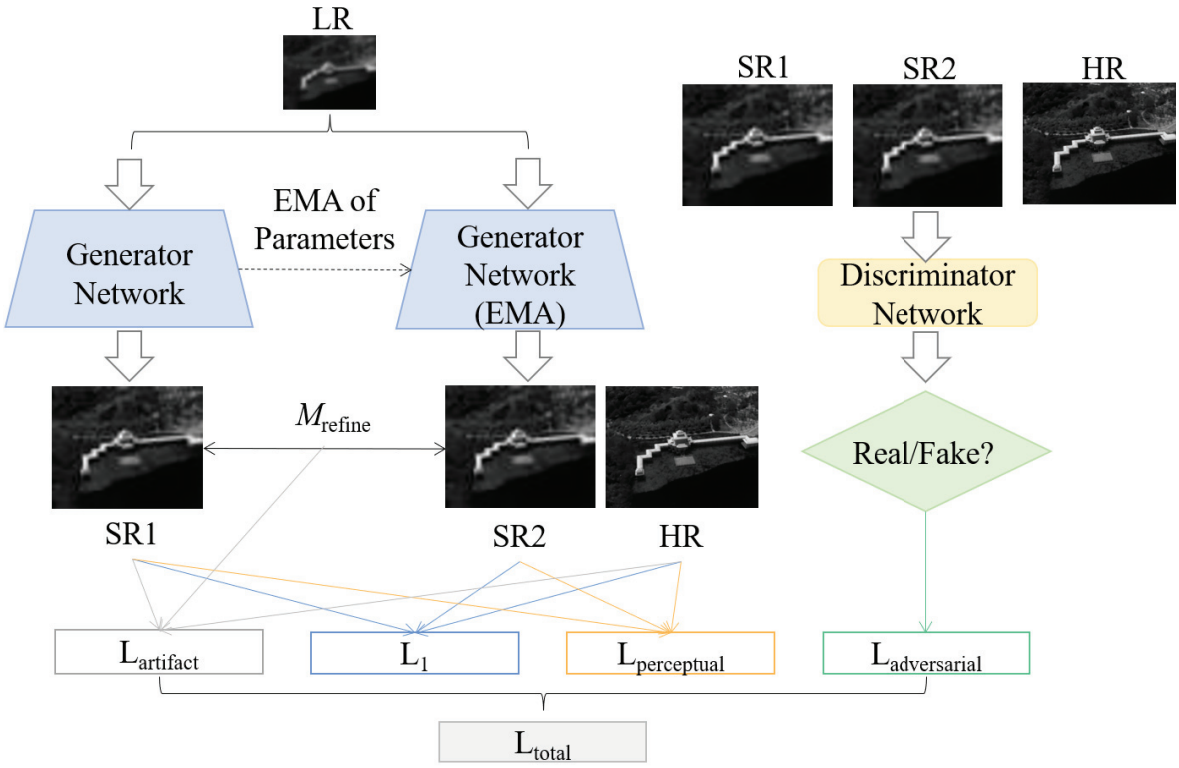


Figure 3. SwinAIR-GAN method. The generator network generates the SR infrared feature map SR1 from the LR infrared image and applies an exponential moving average (EMA) to the parameters to obtain the infrared feature map SR2. Various loss functions are then computed for SR1, SR2, and the HR infrared image. Finally, the corresponding discriminator network judge the authenticity of the generated infrared image.

3.2.1. Generator Network

The generator network utilizes our proposed SwinAIR, which effectively extracts and reconstructs information from infrared images. Detailed information about SwinAIR can be found in Section 3.1.

3.2.2. Discriminator Network

The discriminator network (as shown in Figure 4) employs a U-Net architecture. However, the complex degradation space and structure can increase training instability.

To mitigate this risk, spectral normalization [50] is used to stabilize training. This approach can also alleviate the introduction of unnecessary artifacts during GAN training, thereby achieving a balance between local image detail enhancement and artifact suppression.

The discriminator network primarily comprises 10 convolutional layers, 8 spectral normalization layers, and 3 bilinear interpolation layers. Each of the first 9 convolutional layers is followed by a Leaky ReLU activation function. Table 1 provides the detailed information about each convolutional layer in the discriminator network.

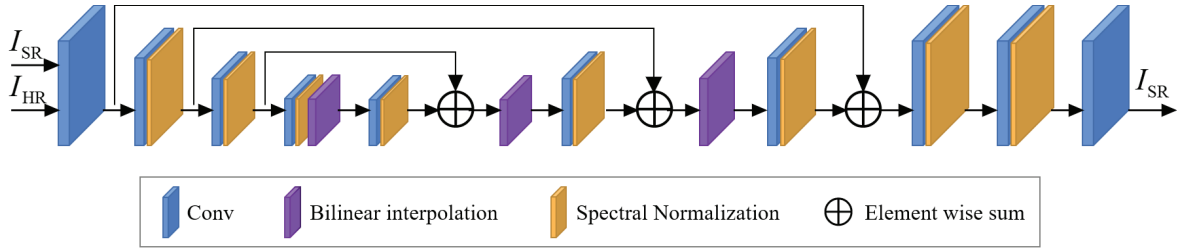


Figure 4. Structure of the discriminator network. We adopt a U-Net structure based on spectral normalization.

Table 1. Detailed information about the convolutional layers in the discriminator network.

Name	Input Channels	Output Channels	Kernel Size	Stride
Conv-1	3	64	3	1
Conv-2	64	128	3	2
Conv-3	128	256	3	2
Conv-4	256	512	3	2
Conv-5	512	256	3	1
Conv-6	256	128	3	1
Conv-7	128	64	3	1
Conv-8	64	64	3	1
Conv-9	64	64	3	1
Conv-10	64	1	3	1

3.2.3. Loss Function

Appropriate loss functions are crucial to quantifying the differences between SR and HR images and optimizing network training. Conventional networks use a combination of per-pixel, perceptual, and adversarial losses in real-image SR tasks. To enhance image details and eliminate artifacts, the SwinAIR-GAN incorporates additional artifact discrimination loss. Below, we will discuss these different loss functions in more detail.

L1 Loss. The information about L1 loss can be found in Section 3.1.3.

Perceptual Loss. In contrast to L1 loss, perceptual loss aims to make SR infrared images more visually similar to HR infrared images rather than just minimizing pixel differences. This process is expressed as follows:

$$L_{\text{perceptual}} = \frac{1}{h_l w_l c_l} \sqrt{\sum_{i,j,k} \left(\phi^l_{i,j,k}(I_{\text{SR}}) - \phi^l_{i,j,k}(I_{\text{HR}}) \right)^2} \tag{15}$$

where h_l, w_l, and c_l denote the height, width, and channel numbers of the lth feature map, respectively. $\phi^l_{i,j,k}(I_{\text{SR}})$ and $\phi^l_{i,j,k}(I_{\text{HR}})$ represent the pixel values of the SR and HR feature maps extracted from the lth layer of the pretrained network at the spatial location (i, j, k), respectively.

Adversarial Loss. Adversarial loss is primarily used to generate realistic SR images. The adversarial losses of SwinAIR-GAN are represented as follows:

$$L_{\text{adversarial_G}} = -\log D_\theta(I_{\text{SR}}) \tag{16}$$

$$L_{\text{adversarial_D}} = -\log D_\theta(I_{\text{HR}}) - \log(1 - D_\theta(I_{\text{SR}})) \tag{17}$$

where $L_{\text{adversarial_G}}$ and $L_{\text{adversarial_D}}$ represent the adversarial losses of the generator network and discriminator network, respectively. $D_\theta(I_{\text{SR}})$ and $D_\theta(I_{\text{HR}})$ denote the probabilities the SR and HR images are trained to be classified as real images, respectively.

Artifact Discrimination Loss. Artifact discrimination loss is used to minimize SR image artifacts [51] as follows:

$$L_{\text{artifact}} = \|M_{\text{refine}} \cdot (I_{\text{HR}} - I_{\text{SR1}})\|_1 \tag{18}$$

where I_{SR1} refers to the SR image obtained through the generator network, and M_{refine} represents the refined feature map.

Overall loss function. Taking the aforementioned losses into account, the comprehensive loss function of the SwinAIR-GAN is as follows:

$$L_{\text{total}} = \alpha \cdot L_1 + \beta \cdot L_{\text{perceptual}} + \gamma \cdot L_{\text{adversarial}} + \delta \cdot L_{\text{artifact}} \tag{19}$$

where α, β and γ represent the proportionate contributions of each loss type (i.e., L_1, perceptual, adversarial and artifact discrimination types) to the comprehensive loss function. Based on experience, we set α=1, β=1, γ=0.1 and δ=1.

3.2.4. Degradation Model

Classical SR networks commonly employed bicubic downsampling to simulate image degradation. However, this approach has difficulty dealing with complex and unknown degradations in real infrared images. Issues such as blurring and ghosting are commonplace. To tackle this problem, we extend the degradation space (as shown in Figure 5) to more effectively simulate the process real infrared image degradation.

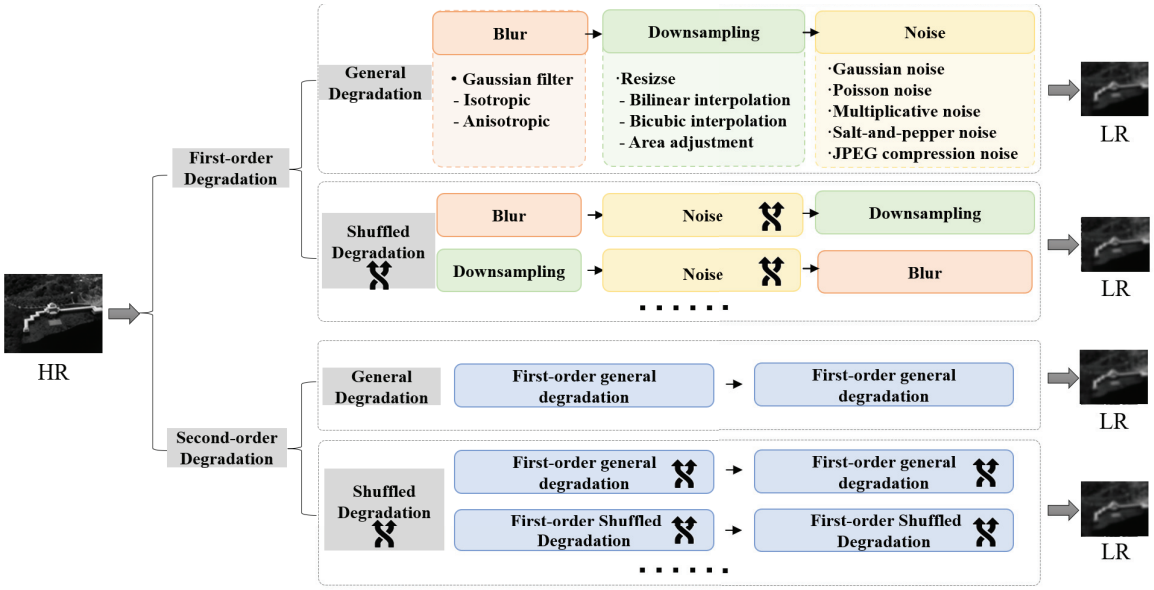


Figure 5. The degradation model. We expand the degradation space and consider first- and second-order degradation processes to emulate the real degradation of infrared images more accurately.

Typically, image degradation can be represented by a general model, such as the following:

$$I_{\text{LR}} = (I_{\text{HR}} \otimes k)\downarrow_s + n_\delta, \{k, s, \delta\} \in R \tag{20}$$

where k denotes the blur kernel, $\otimes$ denotes the convolution operation, $\downarrow$ denotes the downsampling operation, s is the scaling factor, and n_δ represents noise (typically additive white Gaussian) with a standard deviation of δ.

To expand the degradation space, we consider both first- and second-order degradations. For the first-order degenerate process, general and shuffled degradations are applied. The general degradation refers to the sequential application of blur, downsampling, and noise in sets of two or three. The shuffled degradation involves the random ordering of the degradation steps. For the second-order degenerate process, the same two degradation types are considered. The general degradation comprises two first-order degradation processes, while the shuffled degradation involves randomly shuffling blur, downsampling, and noise for each of the two first-order degradation processes.

In the degradation model, blur, noise, and downsampling are the three key factors affecting image degradation. By expanding the degradation space of these factors, true degradation processes are approximated to improve the applicability and generalizability of the network. Our proposed degradation model considers these key factors that affect the degradation of infrared images as follows:

Blur. In our model, isotropic and anisotropic Gaussian blur kernels are used. A Gaussian blur kernel $k(i, j)$ with a kernel size of $2t + 1$ can be represented as follows:

$$k(i,j) = \frac{1}{N}\exp\left(-\frac{1}{2}C^T\sum{}^{-1}C\right), C = [i,j]^T \tag{21}$$

where $\sum$ denotes the covariance matrix, C represents the spatial coordinates, and N is a normalization constant, $(i, j) \in [-t, t]$, and is sampled from a Gaussian distribution. $\sum$ can be further expressed as

$$\sum = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\mu_1^2 & 0\\ 0 & \mu_2^2\end{bmatrix}\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix} \tag{22}$$

where μ_1 and μ_2 represent the standard deviation along two main axes (i.e., the eigenvalues of the covariance matrix), and θ is the rotation angle. When $\mu_1 = \mu_2$, $k(i, j)$ is an isotropic Gaussian blur kernel. Otherwise, $k(i, j)$ is an anisotropic Gaussian blur kernel.

The Gaussian blur kernel configurations are hyperparameters. Different settings affect the blurring of infrared images during degradation and the complexity of the model. Based on previous works (e.g., BSRGAN [13] and Real-ESRGAN [14]) and the characteristics of infrared images, we experiment with various settings combinations and ultimately determine the optimal settings.

For the kernel size, we uniformly sample from $[7 \times 7, 9 \times 9, \cdots, 21 \times 21]$ to obtain varying degrees of blurring. For the isotropic Gaussian kernel, we uniformly sample the standard deviation from $[0.1, 2.8]$. For the anisotropic Gaussian kernel, we uniformly sample the standard deviation from $[0.5, 6]$ and $[0.5, 8]$ with equal probability, and we uniformly sample the rotation angle from $[0, \pi]$ to simulate motion blurring at different angles.

By randomly using these Gaussian kernels with different configurations during the degradation, we achieve a variety of complex blurring effects that closely simulate the degradation process of real infrared images, thus effectively optimizing the performance of SwinAIR-GAN.

Noise. The noise in infrared images can usually be classified into two categories: random noise and fixed pattern noise. Random noise includes two types of stationary random white noise: Gaussian noise and Poisson noise. Fixed pattern noise is primarily characterized by image non-uniformity and blind elements. Non-uniform and scanning-array blind elements contribute to multiplicative noise, whereas staring-array blind elements result in salt-and-pepper noise. Therefore, we consider four types of noise for processing: Gaussian noise, Poisson noise, multiplicative noise, and salt-and-pepper noise. We also include JPEG compression noise to expand the degradation space.

Gaussian noise is random noise with a probability density function (PDF) that follows a Gaussian distribution. The PDF of Gaussian noise is given by

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2} \tag{23}$$

where x represents the grayscale value of the image noise, μ represents the mean value of x, and σ represents the standard deviation of x.

Poisson noise follows a Poisson distribution, and its intensity is proportional to that of the image. The noises at different pixel points are independent of one another. The PDF of Poisson noise is given by

$$p(x = m) = \frac{\lambda^m e^{-\lambda}}{m!}, m = 0, 1, 2, \cdots \tag{24}$$

where λ is the expectant value, and $\lambda > 0$.

Multiplicative noise is produced by random variations in channel characteristics and is related to the signal through multiplication. This type of noise follows a rayleigh or gamma distribution. In infrared images, the multiplicative noise caused by the non-uniformity is numerous and mixed together in a unified form. In a focal plane detector array, the response of each detection element to a mixture of various non-uniform factors can be expressed as follows:

$$x_i = a_i \times I + b_i, i = 1, 2, \cdots, M \tag{25}$$

where x represents the response of the detection element, I represents the incident infrared radiation, and M represents the number of detection elements in the array. The non-uniformity of each detection element is influenced by the combined impact of the gain a_i, and offset, b_i, of each unit.

Salt-and-pepper noise is caused by changes in signal pulse intensity. It includes two variations: high-intensity salt-like noise represented by white dots and low-intensity pepper-like noise represented by black dots. Typically, they both occur simultaneously. Dieickx et al. [52] suggested that the blind elements of a staring array manifest as salt-and-pepper noise. The PDF of salt-and-pepper noise in infrared images can be represented as follows:

$$P(x) = \begin{Bmatrix} P_a, x = a \\ P_b, x = b \\ 0, \text{other} \end{Bmatrix} \tag{26}$$

where P_a and P_b represent the density distributions of pepper noise and salt noise, respectively.

JPEG compression is a widely used lossy compression technique for digital images, which reduces storage and transmission requirements by discarding unimportant components. The quality of the compressed image is determined by the quality factor q ($q \in [0, 100]$). A higher q value corresponds to a lower compression ratio and better image quality. When $q > 90$, the image quality is considered great, and there will be no obvious artifacts introduced. In our degradation model, the range for the JPEG quality factor is set to within $[30, 95]$.

Downsampling. Downsampling is a fundamental operation used to change the image size and generate LR images. There are four interpolation methods for image resizing: nearest neighbor, bilinear, bicubic, and area adjustment. Nearest neighbor interpolation introduces pixel deviation issues [53]; hence, we focus primarily on the other three.

The different resizing methods have their own effects on images. Our degradation model randomly selects from downsampling, upsampling, and maintaining the original size to diversify the degradation space. Assuming that the scaling factor is s, the image size is first changed by a ($a \in [1/2, s]$), resulting in an intermediate-sized image. After degradation, the image is resized by a scaling factor of s/a, causing the image size to match the downsampling scope required by the original image.

4. Experimental Details and Results

In this section, we provide a detailed introduction to the experimental details and validation results of SwinAIR and SwinAIR-GAN. Both quantitative and qualitative analyses demonstrate the superiority of our proposed method for infrared image SR reconstruction.

4.1. Datasets and Metrics

Training sets. Due to the scarcity of high-quality public infrared image datasets available for training, we utilize visible image datasets as the training set based on previous work [45]. Specifically, we use the DV2K [54] (DIV2K [55] + Flickr2K [7]) dataset, which contains 3450 images rich in texture and detail.

Test sets. To maximize the generalization of test scenarios and thoroughly validate the effectiveness of the proposed method, we employ six datasets as the test sets for SwinAIR: the CVC-14 dataset [56], the Flir dataset [57], the Iray universal infrared dataset [58] (Iray-384), the Iray infrared maritime ship dataset [59] (Iray-ship), the Iray infrared aerial photography dataset [60] (Iray-aerial photography), and the Iray infrared security dataset [61] (Iray-security). The CVC-14 dataset [56], the Flir dataset [57], and the Iray-384 dataset [58] mainly consist of road traffic scene images, while the Iray-ship dataset [59], the Iray-aerial photography dataset [60], and the Iray-security dataset [61] consist of maritime ship scenes, aerial street scenes, and security surveillance scenes, respectively. For SwinAIR-GAN, we use the ASL-TID dataset [57], the Iray-384 dataset [58] and a self-built dataset as the test sets. The ASL-TID dataset [57] primarily includes scenes with pedestrians, cats and horses, and our self-built dataset comprises 138 images covering various scenes such as buildings, roads, vehicles and pedestrians.

The purpose of creating our own dataset is to address the challenge of lacking reference images in real infrared image SR tasks, which makes it difficult to thoroughly compare the texture and detail recovery effects of different methods. Specifically, we build an infrared data acquisition system consisting of an LR Iray infrared camera and two HR Chengdu Jinglin infrared cameras. The parameters and models of the cameras are detailed in Table 2. The specific process of acquiring infrared image data using the above-mentioned system is shown in Figure 6. We place three cameras parallel to each other with the LR Iray infrared camera in the middle and the two HR Chengdu Jinglin infrared cameras on the sides. Different degradation methods are applied to infrared images captured by the Iray camera across various scenes to construct LR–HR image pairs (in the custom-degraded SR task, the infrared images captured by the Iray camera are considered as HR images relative to the degraded images). Simultaneously, we perform parallax correction on the images captured by the two Chengdu Jinglin infrared cameras and then stitch them together to obtain images with a higher resolution than the original ones captured by the Iray camera. These higher-resolution images will be used as reference images in real infrared SR tasks (i.e., direct SR reconstruction of images captured by the Iray camera) to evaluate the effectiveness of different methods in reconstructing details and textures from unknown degraded infrared images.

Table 2. Detailed parameters of the infrared data acquisition system.

Camera	Model	Resolution	Focal Length	Horizontal Field of View	Vertical Field of View
Iray	M3384012Y01312X	384 × 288	13mm	20°	15°
Jing Lin Chengdu	RTD6122C	640 × 512	18mm	14.8°	10.8°

Evaluation metrics. To quantitatively evaluate the SR performance of our proposed method, we use full-reference image quality evaluation metrics, peak signal-to-noise ratio (PSNR), and structural similarity (SSIM) for SwinAIR. To further evaluate the realism of the generated infrared images of SwinAIR-GAN, we also use non-reference image quality evaluation metrics, natural image quality evaluator (NIQE), and perceptual index (PI), in addition to PSNR and SSIM. All evaluation metrics are computed on the luminance channel after converting the reconstructed images to the YCbCr color space. We assess the efficiency of SwinAIR using floating-point operations (FLOPs) and inference speed.

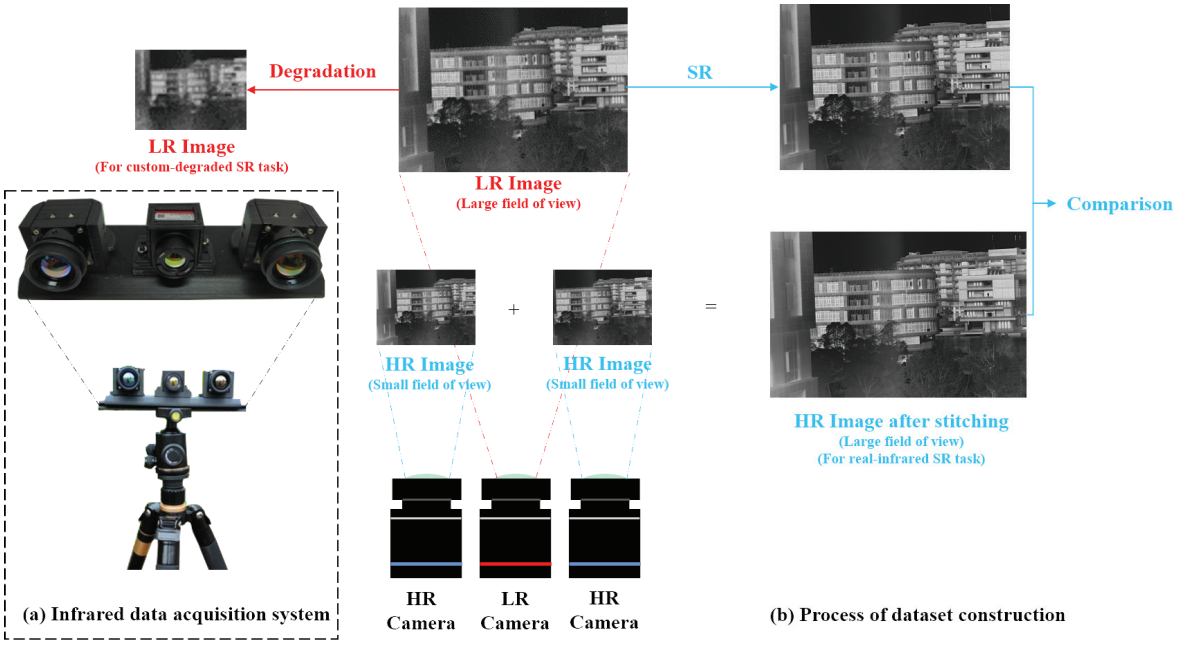


Figure 6. Our infrared data acquisition system. (**a**) System hardware. (**b**) Process of dataset construction. For SR tasks with custom degradation, images captured by the Iray camera are used as HR reference images and to calculate metrics. Due to differences in images captured by different cameras, we use no-reference metrics for the quantitative analysis of SR reconstruction quality for real SR tasks. The HR images captured by the Jinglin Chengdu cameras are stitched together to serve as only reference images for real SR tasks, assessing whether the SR reconstructed images are consistent with real textures and details.

4.2. Model and Training Settings

Model settings. For SwinAIR, the number of RSTABs (n) and the number of STALs (m) are set to 6. The window size of the SW-MSA in STL is 8 with the number of attention heads set to 6. The weights for the high-frequency channel (F_h) and the low-frequency channel (F_l) are set to 1/3 and 2/3, respectively. For SwinAIR-GAN, the degradation model parameters are detailed in Section 3.2.4. The generator network parameters are the same as those of SwinAIR, and the discriminator network settings are provided in Table 1.

Training settings. We use the Adam method to optimize our SwinAIR and SwinAIR-GAN, with the three hyperparameters of Adam, β_1, β_2, and ε, set to 0.9, 0.999, and 10^{-8}, respectively. The initial learning rate is set to 2×10^{-4} and decays using a multi-step strategy. The number of iterations for model updates is set to 6×10^5. All trainings are implemented in PyTorch and conducted on a server equipped with 8 Nvidia A100 GPUs with a batch size of 32. The input LR image size is set to 64 × 64. During the training process, the dataset is augmented using seven data augmentation techniques, including random horizontal flipping, random vertical flipping, and random 90° rotations.

4.3. SwinAIR

4.3.1. Comparisons with the State-of-the-Art Methods

We compare our SwinAIR method with existing mainstream classical image SR methods on multiple datasets to evaluate their performance in reconstructing infrared images. All LR images are degraded using bicubic downsampling.

Quantitative Evaluations. We evaluate the SR reconstruction performance of SwinAIR and other methods on infrared images across six datasets at different scales. The quantitative comparison results are shown in Table 3. Our network achieves superior reconstruction performance with fewer parameters. At the ×2 and ×3 scales, our method achieves the best metrics across all datasets except for the CVC-14 dataset [56], where the results are second best. At the ×4 scale, our method achieves the best performance across all test sets while

reducing the parameters by 19% compared to SwinIR. We also compare the efficiency of SwinAIR with several well-performing methods at the $\times 4$ scale, and the results are shown in Table 4. The FLOPs and inference speed measurements are performed on an NVIDIA A100 GPU with the input image size set to 64×64. To ensure the accuracy of the inference speed measurement, we first perform 50 warm-up inferences, which are followed by 100 inferences to calculate the average inference time. Compared to CNN-based methods (e.g., EDSR [7] and SRFBN [62]), transformer-based methods (e.g., SwinIR [10], HAT [12] and SwinAIR) achieve better performance at the cost of reduced inference speed. Nonetheless, the use of the shift-window mechanism can effectively reduce their computational complexity. Additionally, due to the implementation of channel separation strategies, SwinAIR achieves the lowest computational complexity among the compared methods.

Table 3. Quantitative comparison results achieved by different methods on the CVC-14 [56], Flir [57], Iray-384 [58], Iray-ship [59], Iray-aerial photography [60], and Iray-security [61] datasets. We use PSNR (dB)/SSIM * as the evaluation metrics. Params represents the parameters of the methods. Bold and underlined numbers indicate the best and second-best results, respectively.

Scale	Method	Params	CVC14	Flir	Iray-384	Iray-Security	Iray-Ship	Iray-Aerial Photography
×2	Bicubic	-	39.57/0.9570	32.23/0.7633	29.47/0.8980	33.09/0.9114	34.42/0.9287	32.37/0.9263
	SRCNN [5]	0.1M	39.93/0.9597	32.60/0.7834	31.06/0.9205	34.21/0.9291	35.17/0.9417	34.36/0.9480
	EDSR [7]	43.0M	42.53/0.9666	33.17/0.7930	32.16/0.9314	35.56/0.9383	36.51/0.9492	35.71/0.9586
	SRMD [63]	15.1M	43.62/0.9726	34.40/0.8193	33.14/0.9363	36.36/0.9439	37.67/0.9539	36.74/0.9618
	SRMDNF [63]	15.1M	43.77/0.9730	34.45/0.8202	33.32/0.9380	36.62/0.9454	37.82/0.9548	36.92/0.9630
	SRFBN [62]	3.5M	43.96/0.9736	34.53/0.8214	33.66/0.9409	37.25/0.9484	**38.10**/0.9560	37.21/0.9643
	SwinIR [10]	11.5M	**44.02**/**0.9737**	34.53/0.8208	33.77/0.9418	37.22/0.9484	37.98/0.9563	37.30/**0.9648**
	HAT [12]	20.8M	42.02/0.9701	33.10/0.7964	32.35/0.9320	36.64/0.9468	36.67/0.9504	36.60/0.9610
	SwinAIR (Ours)	9.2M	43.97/**0.9737**	**34.55**/**0.8221**	**33.81**/**0.9420**	**37.40**/**0.9488**	**38.10**/**0.9568**	**37.34**/**0.9648**
×3	Bicubic	-	36.68/0.9274	30.43/0.6947	27.03/0.8324	30.82/0.8575	31.68/0.8807	29.32/0.8596
	SRCNN [5]	0.1M	37.59/0.9352	30.92/0.7172	28.10/0.8601	31.72/0.8803	32.51/0.8982	30.76/0.8911
	EDSR [7]	43.0M	39.29/0.9444	31.45/0.7286	28.85/0.8774	32.57/0.8943	33.68/0.9090	31.78/0.9110
	SRMD [63]	15.3M	40.30/0.9530	32.63/0.7626	29.97/0.8840	33.61/0.9043	34.79/0.9157	32.95/0.9167
	SRMDNF [63]	15.3M	40.43/0.9535	32.69/0.7637	30.06/0.8859	33.70/0.9054	34.87/0.9167	33.04/0.9181
	SRFBN [62]	3.5M	40.81/0.9559	32.84/0.7662	30.38/0.8925	34.06/0.9099	**35.18**/0.9206	33.36/0.9228
	SwinIR [10]	11.5M	**40.87**/**0.9562**	32.84/0.7669	30.42/0.8932	33.97/0.9100	35.15/0.9203	33.37/0.9230
	HAT [12]	20.8M	39.94/0.9442	31.59/0.7285	29.50/0.8779	32.91/0.8974	33.94/0.9103	32.45/0.9146
	SwinAIR (Ours)	9.3M	40.85/0.9561	**32.85**/**0.7673**	**30.49**/**0.8939**	**34.15**/**0.9108**	**35.18**/**0.9212**	**33.38**/**0.9233**
×4	Bicubic	-	34.34/0.8985	29.19/0.6432	25.71/0.7817	29.83/0.8213	30.22/0.8463	27.70/0.8060
	SRCNN [5]	0.1M	35.09/0.9049	29.62/0.6645	26.50/0.8069	30.36/0.8409	30.84/0.8620	28.76/0.8358
	EDSR [7]	43.0M	36.84/0.9212	30.18/0.6813	27.14/0.8280	31.10/0.8592	31.20/0.8754	29.74/0.8642
	SRMD [63]	15.5M	37.87/0.9322	31.38/0.7199	28.29/0.8372	32.20/0.8725	33.20/0.8844	30.89/0.8725
	SRMDNF [63]	15.5M	37.98/0.9333	31.40/0.7210	28.36/0.8405	32.34/0.8750	33.21/0.8859	30.99/0.8755
	SRFBN [62]	3.5M	38.34/0.9367	31.57/0.7249	28.63/0.8483	32.55/0.8792	33.46/0.8905	31.24/0.8812
	SwinIR [10]	11.5M	38.38/0.9372	31.54/0.7251	28.58/0.8482	32.52/0.8794	33.46/0.8907	31.22/0.8815
	HAT [12]	20.8M	37.39/0.9173	30.02/0.6986	27.47/0.8283	31.49/0.8635	32.01/0.8811	30.61/0.8720
	SwinAIR (Ours)	9.3M	**38.52**/**0.9380**	**31.64**/**0.7272**	**28.78**/**0.8512**	**32.66**/**0.8811**	**33.51**/**0.8917**	**31.33**/**0.8835**

* Higher metrics mean better performance.

Table 4. Efficiency comparison results achieved by different methods. FLOPs (G) and inference speed (ms) * are used as metrics for evaluating computational complexity and real-time performance, respectively.

Scale	Image Size	Method	FLOPs (G)	Inference Speed (ms)
×4	64 × 64	EDSR [7]	205.83	10
		SRFBN [62]	530.93	22
		SwinIR [10]	49.28	32
		HAT [12]	82.04	77
		SwinAIR (Ours)	37.51	42

* Lower metrics mean better performance.

Visual comparison. We compare the visual reconstruction results of different methods at the ×4 scale on different datasets. Compared with other methods, our proposed SwinAIR demonstrates superior visual effects and is able to better restore accurate details and textures.

For instance, in the restoration of fence details (Figures 7 and 8), bicubic, SRCNN [5], and EDSR [7] produce visually less appealing blurry images. HAT [12] generates overly sharpened images with speckled and serrated textures. SwinIR [10] and SwinAIR perform better overall, but SwinIR [10] introduces distortions and twists at intersections of distant fence lines. Only SwinAIR can reconstruct such detailed features consistently with reference images. In the restoration of ground textures (Figure 9), all methods except SwinAIR fail to accurately restore the closely spaced horizontal lines at the bottom. Similar situations can also be observed in the contours of cars and ships (Figures 10 and 11), and the lines of door frames (Figure 12), where SwinAIR shows superior capability in recovering fine and contour features. This demonstrates that handling different frequency features separately can effectively enhance the network's performance.

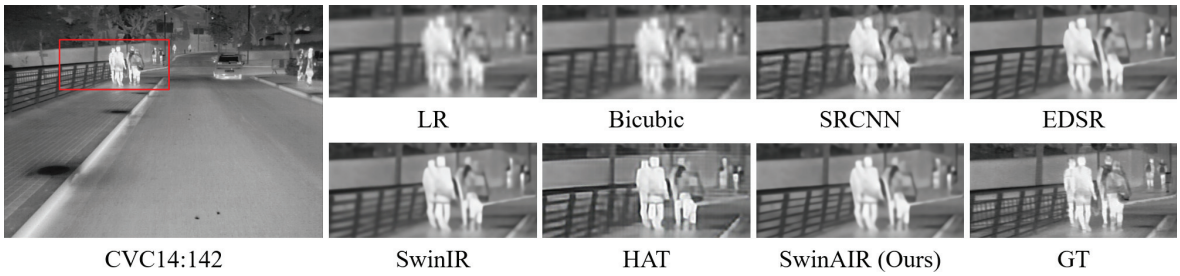


Figure 7. Visual comparison results achieved by different methods on the CVC14 dataset [56] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore more realistic fence details.

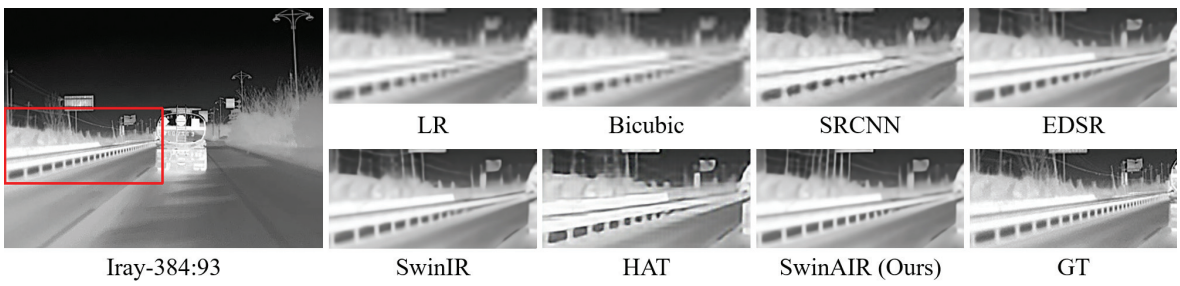


Figure 8. Visual comparison results achieved by different methods on the Iray-384 dataset [58] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore more realistic fence details.

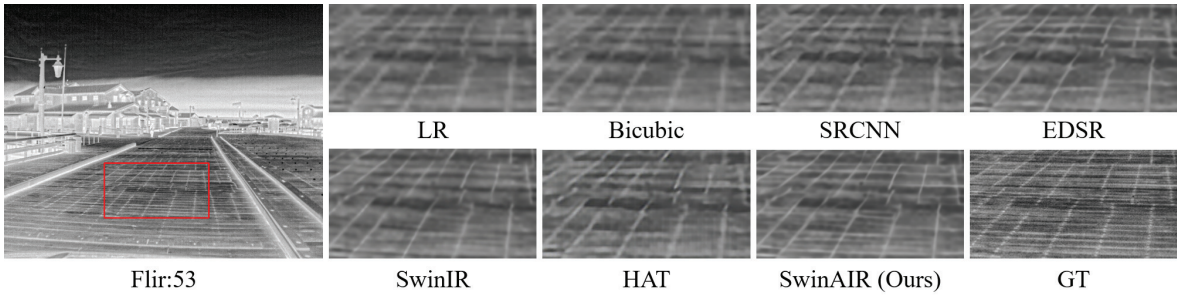


Figure 9. Visual comparison results achieved by different methods on the Flir dataset [57] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore closely spaced ground textures.

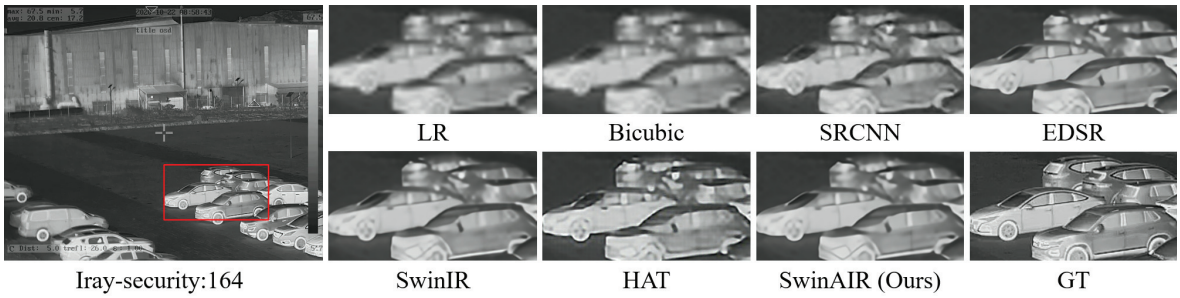


Figure 10. Visual comparison results achieved by different methods on the Iray-security dataset [61] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore more realistic vehicle contours.

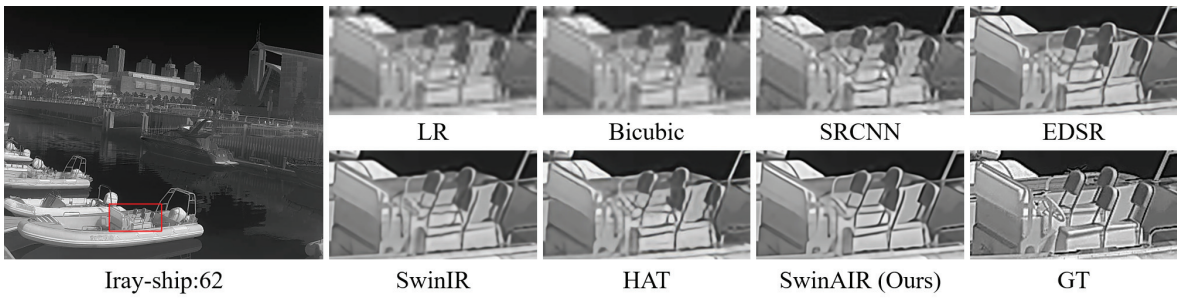


Figure 11. Visual comparison results achieved by different methods on the Iray-ship dataset [59] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore more realistic ship contours.

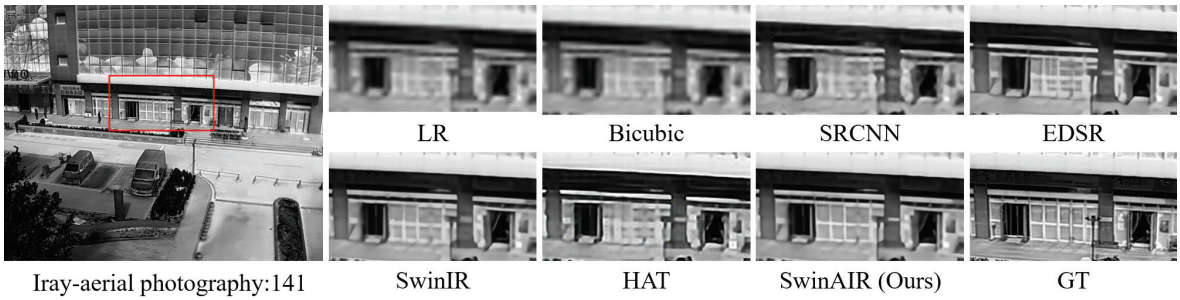


Figure 12. Visual comparison results achieved by different methods on the Iray-aerial photography dataset [60] at the ×4 scale. GT represents the original HR image in the red box of the leftmost image. Our proposed method can restore more realistic lines of door frames.

4.3.2. Ablation Study

To further validate the effectiveness of each component in SwinAIR and the rationality of parameter selection, we conduct extensive ablation experiments. The models are retrained on the DV2K [54] dataset and reevaluated on the CVC14 dataset [56] at the ×4 scale.

Number of RSTABs. Initially, as the number of RSTABs increases, the network's ability to capture features continuously improves, and its performance steadily enhances. However, when the number of RSTABs increases further, the network performance gains begin to saturate while parameters continue to increase rapidly. The relation between network performance and parameters with varying RSTABs is shown in Figure 13a. To balance the performance and parameters of SwinAIR, we ultimately set the number of RSTABs to 6.

Number of STALs in RSTAB. As shown in Figure 13b, the performance of SwinAIR starts to plateau when the number of STALs reaches 4 and saturates when the number reaches 6. There is little difference in performance between having 6 and 8 STALs. Therefore, we ultimately set the number of STALs to 6.

Number of attention heads. Using more attention heads can enhance the network's ability to capture complex features, but it also introduces additional computational overhead. By analyzing the impact of different numbers of attention heads on network performance, as shown in Figure 13c, we find that setting the number of attention heads to 6 achieves a good balance between performance and parameters.

Different weights of the high-frequency and the low-frequency channels. The selection of channel separation weights is related to the distribution of high-frequency and low-frequency information in infrared images. As shown in Figure 13d, we notice that as the weight of the low-frequency channel increases, the network performance continuously improves. This indicates that there is still a substantial amount of learnable low-frequency information in the images. However, due to the high complexity of the low-frequency feature processing structure (STL), the model's parameters also increase accordingly. When the low-frequency channel weight exceeds 2/3, further increasing the weight provides minimal performance gain. Therefore, we ultimately set the high-frequency channel weight to 1/3 and the low-frequency channel weight to 2/3.

Different datasets have varied devices, shooting environments, and objects, which may affect the optimal selection of channel separation weights. To further verify the reasonableness of channel separation weight settings, we perform frequency domain analysis on different datasets. Specifically, we conduct the Fourier transform on the images in each dataset and compute the average spectrum to represent the overall distribution of frequency information in the dataset. The comparison results are shown in Figure 14. Although the spectrograms of different datasets vary in detail, the overall range and intensity of frequency information distribution are similar. This demonstrates that the channel separation weights can be applied to different types of infrared images.

Different structures of the HFEL. Pooling layers in CNNs can reduce data dimensionality and extract high-frequency information from images. We validate the impact of using different pooling layers and different numbers of branches within the HFEL on SwinAIR's performance. The results are shown in Table 5. The results indicate that the average pooling layer outperforms the max pooling layer. Additionally, the structure with two parallel pooling and fully connected layers significantly outperforms the single-branch structure. Interestingly, we find that using both average and max pooling layers together does not perform as well as using either one alone. We hypothesize that this is due to the mismatch and difficulty in fusing features extracted by different pooling layers. Therefore, we ultimately adopt a structure with two parallel average pooling and fully connected layers in the HFEL.

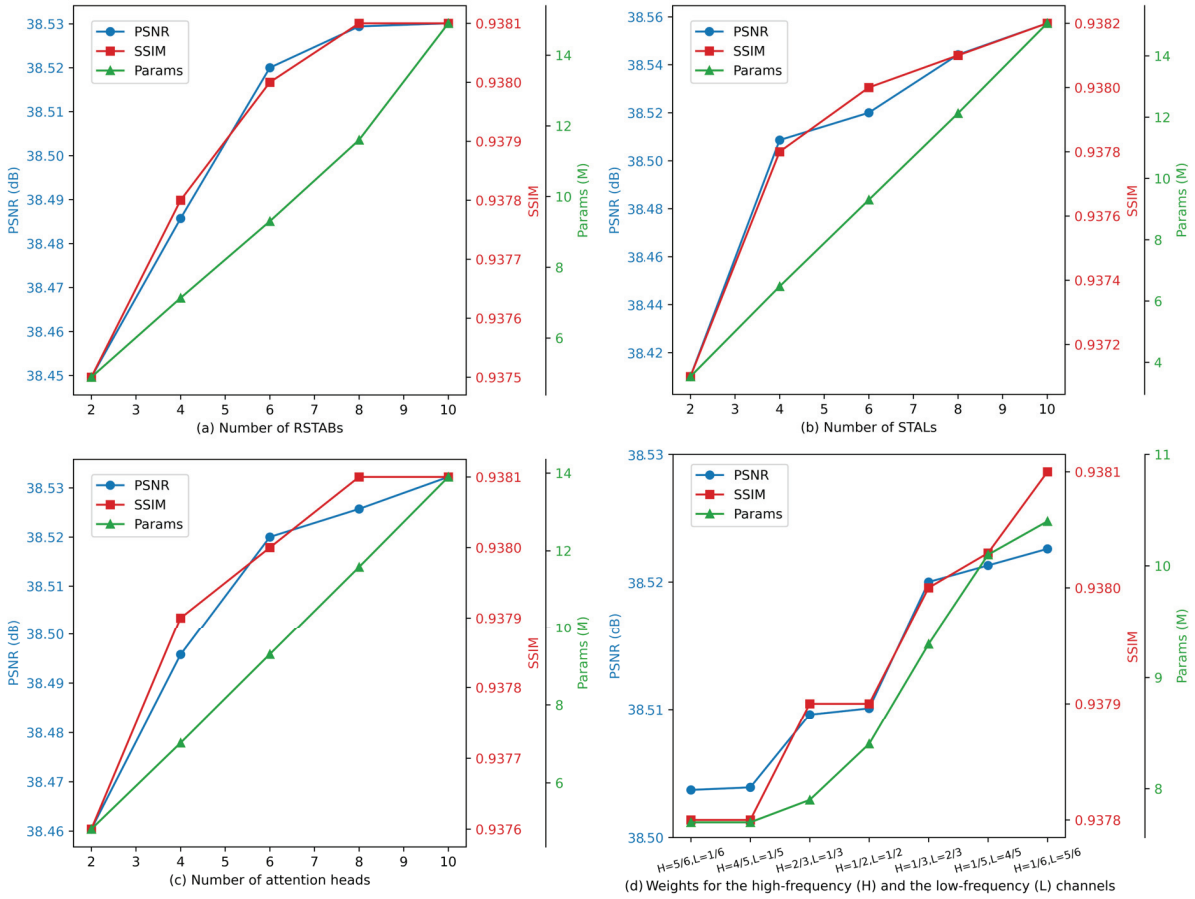


Figure 13. We evaluate the impact of different module configurations and parameter selections on the performance of SwinAIR using PSNR (dB), SSIM, and the number of parameters (M) as performance metrics. In the figures, params represent the number of parameters, H and L denote the weights of high-frequency and low-frequency channels, respectively. (**a**) The impact of the number of RSTABs on the performance of SwinAIR. (**b**) The impact of the number of STALs within each RSTAB on the performance of SwinAIR. (**c**) The impact of the number of attention heads in the STL on the performance of SwinAIR. (**d**) The impact of different weights of the high-frequency and the low-frequency channels on the performance of SwinAIR.

Table 5. Comparison results of using different types of pooling layers and different numbers of branches within the HFEL.

Number of Branches	Avgpool + Linear	Maxpool + Linear	PSNR *(dB)	SSIM *
1	✗	✓	38.19	0.9361
1	✓	✗	38.20	0.9364
2	✓	✗	38.52	0.9380
2	✗	✓	38.51	0.9379
2	✓	✓	37.96	0.9326

* Higher metrics mean better performance.

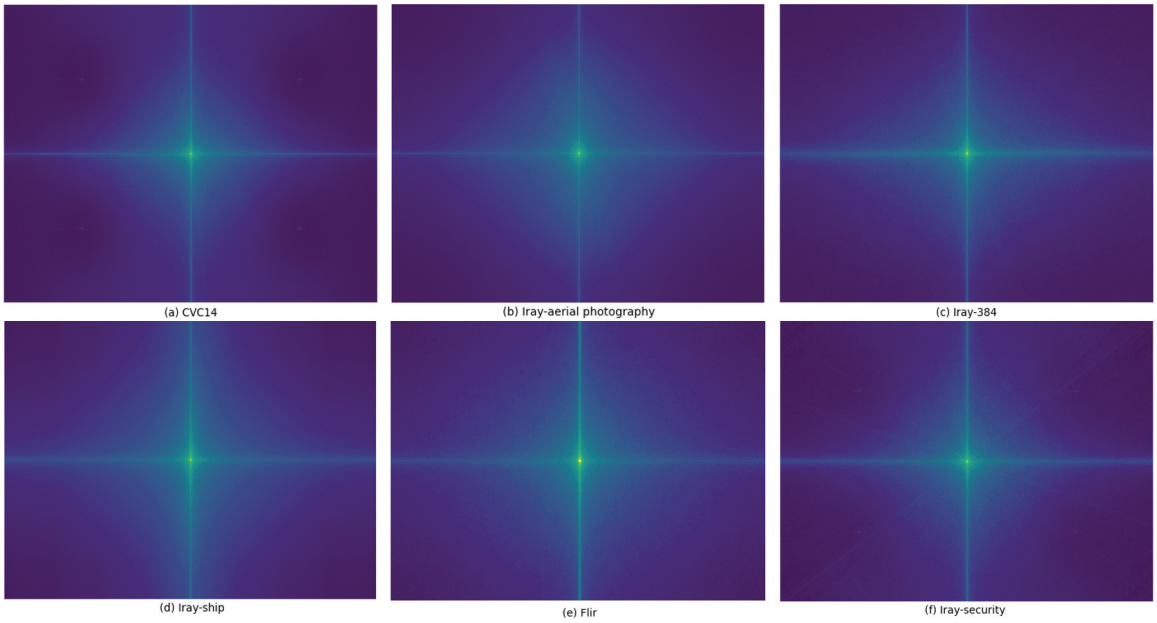


Figure 14. Frequency domain analysis of different datasets. We perform the Fourier transform and spectrum centralization on the images in the datasets. The central part of the spectrogram represents low-frequency information, while the edges represent high-frequency information. The overall range of the spectrum distribution is similar across different datasets.

4.4. SwinAIR-GAN

4.4.1. Comparisons with the State-of-the-Art Methods

We compare the SwinAIR-GAN method with other GAN-based image SR methods across multiple datasets to thoroughly evaluate its performance in restoring real infrared images. For the sake of fairness, we compare methods only with similar types of loss functions due to the strong correlation between evaluation metrics and the loss functions of GAN-based image SR methods [64].

Quantitative Evaluations. We evaluate the SR reconstruction performance of SwinAIR-GAN and other GAN-based methods on infrared images across three datasets. We validate the image restoration performance under both bicubic downsampling (BI degradation) and Gaussian blur downsampling (BD degradation) scenarios at the ×4 scale. The quantitative comparison results are shown in Table 6. Additionally, we perform the ×4 scale SR reconstruction directly on the images in the dataset to validate the SR reconstruction performance of different methods on unknown degraded real scenes. We use no-reference evaluation metrics, NIQE and PI, for quantitative analysis. The quantitative comparison results are shown in Table 7. Apart from achieving second-best results on the ASL-TID dataset [57] for unknown degraded real scenes, SwinAIR-GAN outperforms other models of the same category across all other datasets under various degradation conditions.

Visual comparison. We compare the visual reconstruction results of different methods for real infrared images on multiple datasets at the ×4 scale. Compared with other methods, our proposed SwinAIR-GAN is able to restore details and textures more accurately and achieve better visual effects closer to the real images. For example, in Figure 15, only SwinAIR-GAN restores texture details consistent with the real scene. Other methods either introduce a lot of noise (e.g., ESRGAN [41], RealSR [54]) or generate incorrect details and textures (e.g., Real-ESRGAN [14], BSRGAN [13]). In other scenarios, our method recovers texture details that are more visually accurate and consistent with the LR image (e.g., the restoration of window lines on the left side in Figure 16 and the restoration of the person's arm in Figure 17).

Table 6. Quantitative comparison results achieved by different GAN-based methods on the self-built, Iray-384 [58], and ASL-TID [57] datasets. BI and BD denote the degradation modes using bicubic downsampling and Gaussian blur downsampling, respectively. We use PSNR (dB)/SSIM * as the evaluation metrics. Params represents the parameters of the methods. Bold and underlined numbers indicate the best and second-best results, respectively.

Scale	Method	Loss Function	Self-Built Dataset (BI)	Self-Built Dataset (BD)	Iray-384 (BI)	Iray-384 (BD)	ASL-TID (BI)	ASL-TID (BD)
×4	BSRGAN [13]	$L_1 + L_{\text{perceptual}} + L_{\text{adversarial}}$	31.98/0.8706	31.62/0.8663	25.58/0.7669	25.47/0.7773	33.26/0.8750	32.87/0.8703
	Real-ESRGAN [14]	$L_1 + L_{\text{perceptual}} + L_{\text{adversarial}}$	30.94/0.8630	30.25/0.8513	25.10/0.7681	24.66/0.7703	32.66/0.8787	32.06/0.8704
	SwinIR-GAN [10]	$L_1 + L_{\text{perceptual}} + L_{\text{adversarial}}$	29.91/0.8234	29.67/0.8210	24.64/0.7548	24.59/0.7674	32.87/0.8768	32.53/0.8725
	SwinAIR-GAN (Ours)	$L_1 + 0.1L_{\text{artifact}} + L_{\text{perceptual}} + L_{\text{adversarial}}$	**33.47/0.9106**	**32.93/0.9043**	**25.89/0.7841**	**25.86/0.8037**	**33.82/0.9059**	**33.30/0.8992**

* Higher metrics mean better performance.

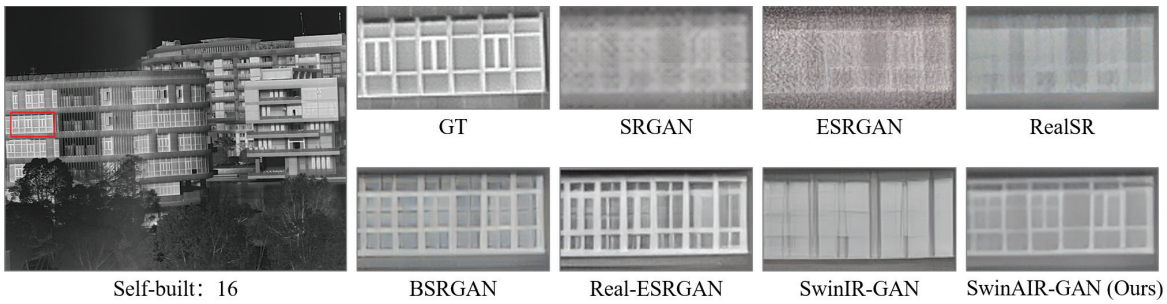


Figure 15. Visual comparison results achieved by different methods for real infrared images on the self-built dataset at the ×4 scale. GT represents the reference image in the red box of the leftmost image, which is obtained by stitching images from two HR cameras. We perform SR reconstruction on the corresponding real-scene images captured by the LR camera and observe the texture differences with the GT images to evaluate performance.

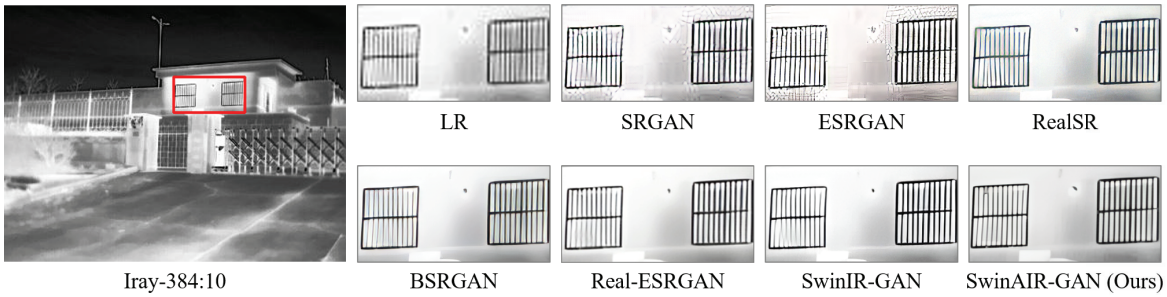


Figure 16. Visual comparison results achieved by different methods for real infrared images on the Iray-384 dataset [58] at the ×4 scale. LR represents the original image in the red box of the leftmost image.

Table 7. Quantitative comparison results achieved by different GAN-based methods on the self-built, Iray-384 [58], and ASL-TID [57] datasets. We perform 4× SR reconstruction on the HR images in the dataset to validate the reconstruction performance of different methods on unknown degradation. NIQE/PI * are used as the evaluation metrics. Params represents the parameters of the methods. Bold and underlined numbers indicate the best and second-best results, respectively.

Scale	Method	Self-Built Dataset	Iray-384	ASL-TID
×4	BSRGAN [13]	6.3307/5.8951	4.3722/4.4544	5.7031/**5.1755**
	Real-ESRGAN [14]	6.2912/6.0504	4.4617/4.3883	6.1816/5.6404
	SwinIR-GAN [10]	6.4183/6.0457	4.0101/4.0049	**5.2140**/**4.9301**
	SwinAIR-GAN (Ours)	**6.0089**/**5.6015**	**3.9406**/**3.9722**	5.4822/4.9886

* Lower metrics mean better performance.

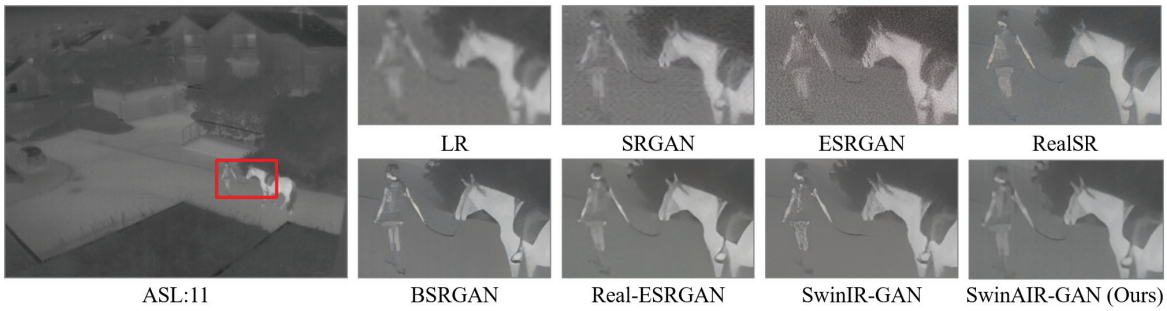


Figure 17. Visual comparison results achieved by different methods for real infrared images on the ASL-TID dataset [57] at the ×4 scale. LR represents the original image in the red box of the leftmost image.

4.4.2. Ablation Study

We conduct further ablation experiments on SwinAIR-GAN to validate the effectiveness of the dropout operation and the artifact discrimination loss. The models are retrained on the DV2K [54] dataset and reevaluated on the self-built dataset at the ×4 scale. We quantitatively assess the experimental results under custom degradation (using PSNR (dB)/SSIM as reference metrics) and unknown degradation (using NIQE/PI as reference metrics).

Effectiveness of the dropout operation. The dropout operation included in the SwinAIR-GAN lead to improved performance across all evaluation metrics compared with the same model without the dropout operation. The specific results can be found in Table 8. For real infrared images, the dropout operation results in an NIQE reduction of 0.2964 and a PI reduction of 0.1173. Furthermore, there are significant improvements in PSNR and SSIM under various degradations.

Table 8. Comparison results of whether use dropout operation or not.

Dropout	Unknown Degradation (NIQE/PI) [1]	BI Degradation (PSNR (dB)/SSIM) [2]	BD Degradation (PSNR (dB)/SSIM) [2]
✗	6.3053/5.7188	32.46/0.8904	32.22/0.8924
✓	6.0089/5.6015	33.47/0.9106	32.93/0.9043

[1] Lower NIQE/PI mean better performance. [2] Higher PSNR (dB) /SSIM mean better performance.

Effectiveness of the artifact discrimination loss. The SwinAIR-GAN incorporating the artifact discrimination loss exhibits lower NIQE/PI scores in the real infrared SR task and

higher PSNR (dB) /SSIM scores in the custom-degraded SR task than the model without. The detailed results are shown in Table 9.

Table 9. Comparison results of whether use artifact discrimination loss or not.

Dropout	Unknown Degradation (NIQE/PI) [1]	BI Degradation (PSNR (dB)/SSIM) [2]	BD Degradation (PSNR (dB)/SSIM) [2]
✗	6.6458/6.0230	32.67/0.8969	32.37/0.8934
✓	6.0089/5.6015	33.47/0.9106	32.93/0.9043

[1] Lower NIQE/PI mean better performance. [2] Higher PSNR (dB) /SSIM mean better performance.

5. Conclusions

In this study, we devise a novel method, SwinAIR, for infrared image SR reconstruction. Through our carefully designed RSTAB, SwinAIR is capable of extracting both low- and high-frequency information from infrared images, enabling effective resolution enhancement. Building on this foundation, we further explore the SR reconstruction task for real infrared images by integrating SwinAIR with the U-Net to create SwinAIR-GAN. SwinAIR-GAN expands the degradation space to better simulate the process of real infrared image degradation. In addition, spectral normalization, dropout operation, and artifact discrimination loss are added to further enhance its performance. To better evaluate the reconstruction effects of different methods on real infrared images, we build an infrared data acquisition system. This system captures corresponding LR and HR infrared images of the same scene, addressing the lack of reference images in real infrared image SR tasks. Extensive comparative experiments on various datasets demonstrate the effectiveness of SwinAIR and SwinAIR-GAN. Although our method achieves superior performance and delivers visually striking results, there may still be differences between the reconstruction images by SwinAIR-GAN and the real infrared images due to the lack of high-quality infrared datasets [65]. Recent studies have shown that transfer learning can effectively improve network performance under the scarcity of datasets [45,66,67]. In the future, we will consider employing this technique, as well as expanding the training dataset, to further explore new methods for the real infrared image SR reconstruction task.

Author Contributions: Methodology, F.H.; Validation, J.W.; Writing—original draft, Y.L.; Writing—review and editing, X.Y.; Supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Henn, K.A.; Peduzzi, A. Surface Heat Monitoring with High-Resolution UAV Thermal Imaging: Assessing Accuracy and Applications in Urban Environments. *Remote Sens.* **2024**, *16*, 930. [CrossRef]
2. Chen, X.; Letu, H.; Shang, H.; Ri, X.; Tang, C.; Ji, D.; Shi, C.; Teng, Y. Rainfall Area Identification Algorithm Based on Himawari-8 Satellite Data and Analysis of its Spatiotemporal Characteristics. *Remote Sens.* **2024**, *16*, 747. [CrossRef]
3. Cheng, L.; He, Y.; Mao, Y.; Liu, Z.; Dang, X.; Dong, Y.; Wu, L. Personnel Detection in Dark Aquatic Environments Based on Infrared Thermal Imaging Technology and an Improved YOLOv5s Model. *Sensors* **2024**, *24*, 3321. [CrossRef]
4. Calvin, W.M.; Littlefield, E.F.; Kratt, C. Remote sensing of geothermal-related minerals for resource exploration in Nevada. *Geothermics* **2015**, *53*, 517–526. [CrossRef]
5. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part IV 13; Springer: Berlin/Heidelberg, Germany, 2014, pp. 184–199.

6. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
7. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
8. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
9. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-trained image processing transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12299–12310.
10. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. Swinir: Image restoration using swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844.
11. Zhang, D.; Huang, F.; Liu, S.; Wang, X.; Jin, Z. Swinfir: Revisiting the swinir with fast fourier convolution and improved training for image super-resolution. *arXiv* **2022**, arXiv:2208.11247.
12. Chen, X.; Wang, X.; Zhou, J.; Qiao, Y.; Dong, C. Activating more pixels in image super-resolution transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22367–22377.
13. Zhang, K.; Liang, J.; Van Gool, L.; Timofte, R. Designing a practical degradation model for deep blind image super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4791–4800.
14. Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1905–1914.
15. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
17. Wang, J.; Ralph, J.F.; Goulermas, J.Y. An analysis of a robust super resolution algorithm for infrared imaging. In Proceedings of the 2009 Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, Salzburg, Austria, 16–18 September 2009; pp. 158–163.
18. Choi, K.; Kim, C.; Kang, M.H.; Ra, J.B. Resolution improvement of infrared images using visible image information. *IEEE Signal Process. Lett.* **2011**, *18*, 611–614. [CrossRef]
19. Mao, Y.; Wang, Y.; Zhou, J.; Jia, H. An infrared image super-resolution reconstruction method based on compressive sensing. *Infrared Phys. Technol.* **2016**, *76*, 735–739. [CrossRef]
20. Deng, C.Z.; Tian, W.; Chen, P.; Wang, S.Q.; Zhu, H.S.; Hu, S.F. Infrared image super-resolution via locality-constrained group sparse model. *Acta Phys. Sin.* **2014**, 63, 044202. [CrossRef]
21. Yang, X.; Wu, W.; Hua, H.; Liu, K. Infrared image recovery from visible image by using multi-scale and multi-view sparse representation. In Proceedings of the 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, Thailand, 23–27 November 2015; pp. 554–559.
22. Yang, X.; Wu, W.; Liu, K.; Zhou, K.; Yan, B. Fast multisensor infrared image super-resolution scheme with multiple regression models. *J. Syst. Archit.* **2016**, *64*, 11–25. [CrossRef]
23. Song, P.; Deng, X.; Mota, J.F.; Deligiannis, N.; Dragotti, P.L.; Rodrigues, M.R. Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries. *IEEE Trans. Comput. Imaging* **2019**, *6*, 57–72. [CrossRef]
24. Yao, T.; Luo, Y.; Hu, J.; Xie, H.; Hu, Q. Infrared image super-resolution via discriminative dictionary and deep residual network. *Infrared Phys. Technol.* **2020**, *107*, 103314. [CrossRef]
25. Wang, Y.; Wang, L.; Liu, B.; Zhao, H. Research on blind super-resolution technology for infrared images of power equipment based on compressed sensing theory. *Sensors* **2021**, *21*, 4109. [CrossRef] [PubMed]
26. Alonso-Fernandez, F.; Farrugia, R.A.; Bigun, J. Iris super-resolution using iterative neighbor embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 153–161.
27. Ahmadi, S.; Burgholzer, P.; Jung, P.; Caire, G.; Ziegler, M. Super resolution laser line scanning thermography. *Opt. Lasers Eng.* **2020**, *134*, 106279. [CrossRef]
28. Wang, Y.; Zhang, J.; Wang, L. Compressed Sensing Super-Resolution Method for Improving the Accuracy of Infrared Diagnosis of Power Equipment. *Appl. Sci.* **2022**, *12*, 4046. [CrossRef]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
30. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
31. Lin, X.; Sun, S.; Huang, W.; Sheng, B.; Li, P.; Feng, D.D. EAPT: Efficient attention pyramid transformer for image processing. *IEEE Trans. Multimed.* **2021**, *25*, 50–61. [CrossRef]

32. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer; Cham, Switzerland, 2020; pp. 213–229.
33. Ma, C.; Zhuo, L.; Li, J.; Zhang, Y.; Zhang, J. Cascade transformer decoder based occluded pedestrian detection with dynamic deformable convolution and gaussian projection channel attention mechanism. *IEEE Trans. Multimed.* **2023**, *25*, 1529–1537. [CrossRef]
34. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. Vivit: A video vision transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6836–6846.
35. Junayed, M.S.; Islam, M.B. Consistent video inpainting using axial attention-based style transformer. *IEEE Trans. Multimed.* **2022**, 25, 7494–7504. [CrossRef]
36. Cao, Y.; Li, L.; Liu, B.; Zhou, W.; Li, Z.; Ni, W. CFMB-T: A cross-frequency multi-branch transformer for low-quality infrared remote sensing image super-resolution. *Infrared Phys. Technol.* **2023**, *133*, 104861. [CrossRef]
37. Yi, S.; Li, L.; Liu, X.; Li, J.; Chen, L. HCTIRdeblur: A hybrid convolution-transformer network for single infrared image deblurring. *Infrared Phys. Technol.* **2023**, *131*, 104640. [CrossRef]
38. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
39. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
40. Yan, B.; Bare, B.; Ma, C.; Li, K.; Tan, W. Deep objective quality assessment driven single image super-resolution. *IEEE Trans. Multimed.* **2019**, *21*, 2957–2971. [CrossRef]
41. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
42. Shang, T.; Dai, Q.; Zhu, S.; Yang, T.; Guo, Y. Perceptual extreme super-resolution network with receptive field block. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 440–441.
43. Liu, S.; Yang, Y.; Li, Q.; Feng, H.; Xu, Z.; Chen, Y.; Liu, L. Infrared image super resolution using gan with infrared image prior. In Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, 19–21 July 2019; pp. 1004–1009.
44. Huang, Y.; Jiang, Z.; Wang, Q.; Jiang, Q.; Pang, G. Infrared image super-resolution via heterogeneous convolutional WGAN. In Proceedings of the PRICAI 2021: Trends in Artificial Intelligence: 18th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2021, Hanoi, Vietnam, 8–12 November 2021; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2021; pp. 461–472.
45. Huang, Y.; Jiang, Z.; Lan, R.; Zhang, S.; Pi, K. Infrared image super-resolution via transfer learning and PSRGAN. *IEEE Signal Process. Lett.* **2021**, *28*, 982–986. [CrossRef]
46. Liu, Q.M.; Jia, R.S.; Liu, Y.B.; Sun, H.B.; Yu, J.Z.; Sun, H.M. Infrared image super-resolution reconstruction by using generative adversarial network with an attention mechanism. *Appl. Intell.* **2021**, *51*, 2018–2030. [CrossRef]
47. Lee, I.H.; Chung, W.Y.; Park, C.G. Style transformation super-resolution GAN for extremely small infrared target image. *Pattern Recognit. Lett.* **2023**, *174*, 1–9. [CrossRef]
48. Kong, X.; Liu, X.; Gu, J.; Qiao, Y.; Dong, C. Reflash dropout in image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6002–6012.
49. Si, C.; Yu, W.; Zhou, P.; Zhou, Y.; Wang, X.; Yan, S. Inception transformer. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23495–23509.
50. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv* **2018**, arXiv:1802.05957.
51. Liang, J.; Zeng, H.; Zhang, L. Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5657–5666.
52. Dierickx, B.; Meynants, G. Missing pixel correction algorithm for image sensors. In *Advanced Focal Plane Arrays and Electronic Cameras II*; SPIE: Bellingham, WA, USA, 1998; Volume 3410, pp. 200–203.
53. Zhang, K.; Gool, L.V.; Timofte, R. Deep unfolding network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3217–3226.
54. Ji, X.; Cao, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F. Real-world super-resolution via kernel estimation and noise injection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 466–467.
55. Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.H.; Zhang, L. Ntire 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 114–125.
56. González, A.; Fang, Z.; Socarras, Y.; Serrat, J.; Vázquez, D.; Xu, J.; López, A.M. Pedestrian detection at day/night time with visible and FIR cameras: A comparison. *Sensors* **2016**, *16*, 820. [CrossRef]

57. Portmann, J.; Lynen, S.; Chli, M.; Siegwart, R. People detection and tracking from aerial thermal views. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1794–1800.
58. Iray-384 Image Database. Available online: http://openai.iraytek.com/apply/Universal_video.html/ (accessed on 21 May 2024).
59. Iray-Ship Image Database. Available online: http://openai.raytrontek.com/apply/Sea_shipping.html/ (accessed on 21 May 2024).
60. Iray-Aerial Photography Image Database. Available online: http://openai.iraytek.com/apply/Aerial_mancar.html/ (accessed on 21 May 2024).
61. Iray-Security Image Database. Available online: http://openai.iraytek.com/apply/Infrared_security.html/ (accessed on 21 May 2024).
62. Li, Z.; Yang, J.; Liu, Z.; Yang, X.; Jeon, G.; Wu, W. Feedback network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3867–3876.
63. Zhang, K.; Zuo, W.; Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3262–3271.
64. Sajjadi, M.S.; Scholkopf, B.; Hirsch, M. Enhancenet: Single image super-resolution through automated texture synthesis. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4491–4500.
65. Liang, S.; Song, K.; Zhao, W.; Li, S.; Yan, Y. DASR: Dual-Attention Transformer for infrared image super-resolution. *Infrared Phys. Technol.* **2023**, *133*, 104837. [CrossRef]
66. Wei, W.; Sun, Y.; Zhang, L.; Nie, J.; Zhang, Y. Boosting one-shot spectral super-resolution using transfer learning. *IEEE Trans. Comput. Imaging* **2020**, *6*, 1459–1470. [CrossRef]
67. Zhang, B.; Ma, M.; Wang, M.; Hong, D.; Yu, L.; Wang, J.; Gong, P.; Huang, X. Enhanced resolution of FY4 remote sensing visible spectrum images utilizing super-resolution and transfer learning techniques. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 7391–7399. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

RetinaViT: Efficient Visual Backbone for Online Video Streams

Tomoyuki Suzuki * and Yoshimitsu Aoki

Department of Electronics and Electrical Engineering, Faculty of Science and Technology, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, Kanagawa, Japan; aoki@elec.keio.ac.jp
* Correspondence: tomoyukun@keio.jp

Abstract: In online video understanding, which has a wide range of real-world applications, inference speed is crucial. Many approaches involve frame-level visual feature extraction, which often represents the biggest bottleneck. We propose RetinaViT, an efficient method for extracting frame-level visual features in an online video stream, aiming to fundamentally enhance the efficiency of online video understanding tasks. RetinaViT is composed of efficiently approximated Transformer blocks that only take changed tokens (event tokens) as queries and reuse the already processed tokens from the previous timestep for the others. Furthermore, we restrict keys and values to the spatial neighborhoods of event tokens to further improve efficiency. RetinaViT involves tuning multiple parameters, which we determine through a multi-step process. During model training, we randomly vary these parameters and then perform black-box optimization to maximize accuracy and efficiency on the pre-trained model. We conducted extensive experiments on various online video recognition tasks, including action recognition, pose estimation, and object segmentation, validating the effectiveness of each component in RetinaViT and demonstrating improvements in the speed/accuracy trade-off compared to baselines. In particular, for action recognition, RetinaViT built on ViT-B16 reduces inference time by approximately 61.9% on the CPU and 50.8% on the GPU, while achieving slight accuracy improvements rather than degradation.

Keywords: online video understanding; Vision Transformer; efficient computation

1. Introduction

Video is continuous data in time; the information obtained from the current frame is less significant when conditioned on previously observed frames than when not. Biovision efficiently recognizes real-world video in directed time by focusing only on areas of change, "events", and reducing redundancies inherent in a video [1]. Inspired by this, efficient sensor devices have been developed that generate signals asynchronously only when luminance changes [2]. In video compression, the most popular approach involves encoding frames in a video with luminance changes and motions from a keyframe [3].

What about online video understanding, where a computer sequentially observes frames in a directed video stream and outputs understandable information? Many methods are based on frame-level visual feature extraction at an equivalent computational cost [4–6]. In online video understanding, which often requires real-time processing, computational efficiency, especially in terms of inference speed, is paramount, and the largest bottleneck involves frame-level visual feature extraction. Considering the redundancy in a video, restricting the process to only the informative areas of the video conditioned on the past has the potential to significantly reduce computational costs.

Although there are many efficient online video understanding methods that account for video redundancy for specific tasks (e.g., action recognition [4,7], pose estimation [5,8]), there are only a limited number of task-agnostic methods. Skipping unchanged parts of the frame image is a simple and reasonable idea, and methods for convolutional neural networks (CNNs) have realized reductions in theoretical computation costs [9,10]. However, the overhead of unstructured memory access due to the naive implementation of

convolution on sparse input has made it difficult to improve actual inference speed, even if the theoretical computational costs were reduced. Recently, a sophisticated CUDA implementation of this skipping idea has improved actual inference speed [11,12].

Meanwhile, Vision Transformer (ViT) [13], which takes an image as a sequence of patches (tokens) and analyzes them with a Transformer [14], derived from natural language processing, has emerged in recent years as the backbone for broad vision tasks [15–17]. ViT can strongly scale with the amount of training data due to its representational capacity, and pre-trained ViTs have demonstrated performance that rivals or surpasses CNNs as the backbone for various vision tasks, making them promising for online video understanding. The Transformer block, the basic element of ViT, operates as a set-to-set transformation. Reducing the number of tokens reduces computation without increasing memory access overhead. Focusing on these properties, we implement visual sparse processing on ViT to improve actual inference speed for online video.

In this study, we propose RetinaViT, which can efficiently extract visual features from online video streams by reducing the redundancy of the target video. We show an overview of RetinaViT in Figure 1. The basic concept of RetinaViT is intuitive: only the changed parts in a frame (event tokens) are processed, and for the other parts, previously processed information is reused. This simple design does not assume any specific task, making RetinaViT a task-agnostic backbone capable of extracting visual features at a wide range of granularities, from low to high levels. Specifically, we approximate the process of the Transformer block by inputting only event tokens as queries and reusing the previous information for the rest of the tokens. In addition, inspired by recently developed local attention methods [18], we further improve efficiency by restricting keys and values to the neighborhood of event tokens. Due to the nature of the transformer block, reducing the number of tokens directly leads to improvements in actual inference speed without requiring a cumbersome implementation. RetinaViT has multiple approximation parameters, and we randomly vary them during model training to create a single model that adapts to any approximation parameter setting. After training, black-box optimization is used to find parameters that make the model both accurate and computationally efficient.

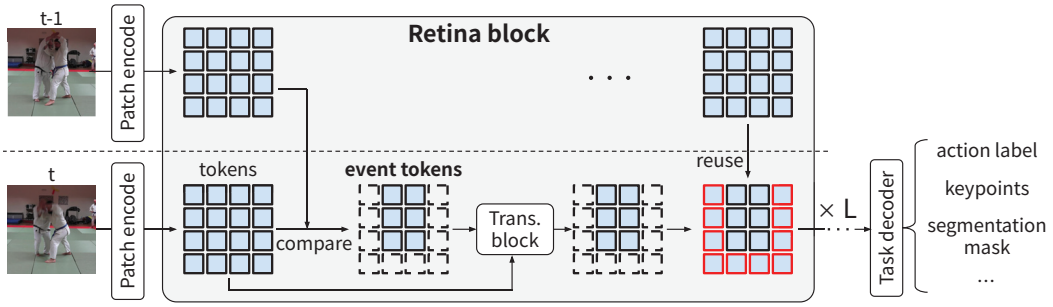


Figure 1. Overview of RetinaViT. Based on Vision Transformer (ViT) [13], RetinaViT converts an input frame image into tokens and processes them with a stack of Transformer blocks. The key difference is that RetinaViT detects tokens that have changed compared to those at the previous timestep in the same stage (block), referred to as event tokens. It then inputs only these event tokens as queries to the Transformer blocks for feature extraction, while reusing the previous tokens for the rest (represented by rectangles with red edges). For simplicity, this figure does not show the restriction of keys and values to the neighborhood of event tokens (see Figure 2 and Section 3.1 for details). This simple framework is task-agnostic, and RetinaViT can be used as the backbone for a wide range of online video recognition tasks.

Our main contribution is the proposal of RetinaViT, and we summarize its features as follows: RetinaViT is (1) a task-agnostic visual feature backbone for online video understanding that (2) leverages the strong representational power of ViTs, and (3) does not require cumbersome implementations such as custom CUDA kernels. RetinaViT is the first method to realize these features simultaneously. In our experiments, we conducted

an extensive ablation study and comparison with existing methods on three online video understanding tasks, each with different characteristics, and demonstrated the effectiveness of our method in improving the trade-off between accuracy and efficiency.

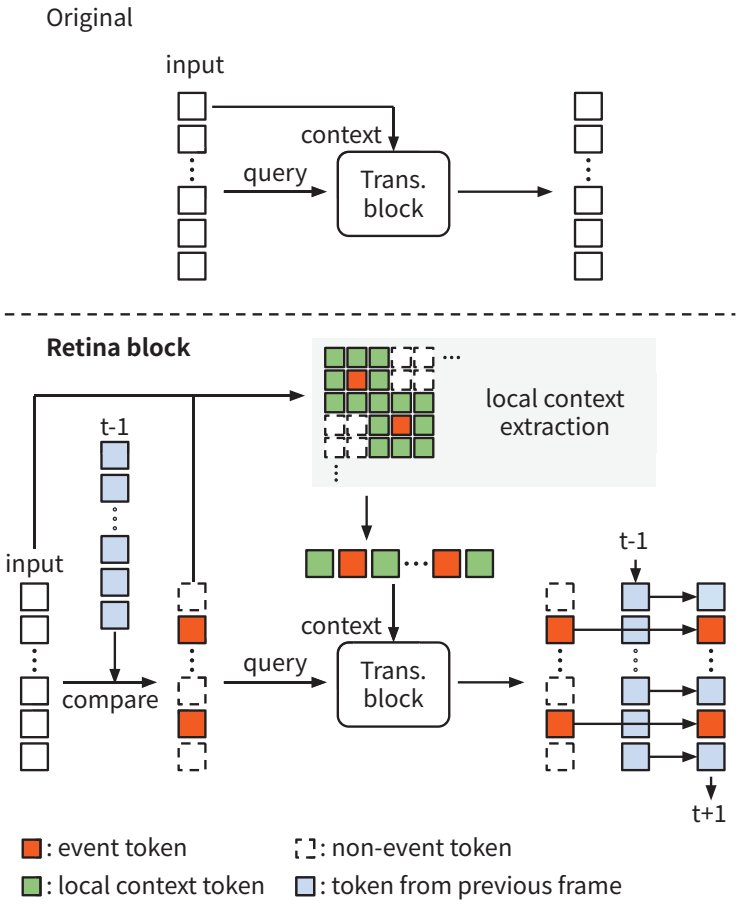


Figure 2. Original Transformer block (**top**) and Retina block (**bottom**). Each rectangle represents a token. In the Retina block, only event tokens, i.e., tokens that have changed over time, are input as queries, and the previous information is reused for the rest. In addition, by restricting the context tokens to the spatial neighborhood of the event tokens, the computational cost is further reduced.

2. Related Works

Online video understanding. Online video understanding is a general term for tasks that involve the observation of frames moment by moment and output understandable information (e.g., action labels in online action recognition and keypoint coordinates in online pose estimation). Processing each frame independently (i.e., using a sliding-window approach) is the most basic method and has proven to be a competitive solution for online video understanding [5,19]. The dominant approach consists of conducting visual feature extraction at each frame using a visual encoder (e.g., CNN, ViT) and a task-specific decoder, often including temporal reasoning for current and past features. For example, in action recognition, ECO [4] and TRN [7] independently extract frame-level features using 2D CNNs and then use 3D CNNs and LSTMs [20] to associate them, respectively. The recently proposed method using Transformers [14] as decoders also requires frame-level visual feature extraction before the decoder [21–24]. In online pose estimation, methods exist to extract frame-level features and relate them using LSTMs [8], warping and graph inference [25], and kernel distillation [26]. The dominant paradigm in online object tracking, tracking-by-detection [20,27–31], assumes frame-by-frame object detection. Therefore, frame-level visual feature extraction is fundamental in online video understanding but also accounts for a large portion of the overall computational cost. Our RetinaViT is an efficient

frame-level feature extractor that has the potential to fundamentally improve the efficiency of many online video understanding tasks in a plug-and-play manner.

Efficient neural visual backbone. As neural visual models have been fundamental for a broad range of computer vision tasks, the development of efficient neural visual models has attracted considerable interest from researchers. One perspective categorizes approaches for improving the efficiency of neural visual models into static or dynamic. While the static approach imposes a constant computational cost regardless of the input, the dynamic approach adapts the computational cost based on the complexity of the input. Optimizing the model structure and neural operations is a common static approach. This includes techniques such as convolution decomposition in CNNs [32–34], local attention [18], and the decomposition of attention [35,36] in ViTs, as well as neural architecture search [37,38]. In addition, pruning [39–41] and quantization [42,43] also fall under the static approach. In contrast, the dynamic approach includes methods that dynamically change the computational graph, such as early termination [44,45], which changes the depth of layers to be processed based on input data. Methods that select blocks to be processed [46], dynamic quantization [47], and patch selection [48–51] also fall under this category. Dynamic approaches tend to be more complex than static approaches, as they require a policy model to make processing decisions separate from the main model [46] or additional loss functions [47]. In principle, however, dynamic approaches have a higher potential for efficiency than static approaches because they can optimize the allocation of processing costs based on the complexity and importance of the input data. Our method falls under the dynamic approach and is unique compared to these methods in that it focuses on redundancies inherent in online videos.

There are methods similar to ours that focus on video redundancy to improve efficiency. In static approaches used for online video pose estimation and online video object segmentation, methods that warp information processed in the past based on motion information reduce computational costs [8,52]. Several dynamic methods adjust computational costs (e.g., model size [53,54], quantization accuracy [55], and input resolution [54]) using a policy model based on past information. In particular, AR-Net [54], which dynamically changes the input resolution, is similar to our proposed method in terms of adaptive input adjustment; however, our method selects image patches, which allows for more granular adjustments. In addition, while all these methods assume specific tasks, our RetinaViT is a task-agnostic, efficient visual feature backbone for online video understanding.

The study of processing convolution only at the point of change is most relevant to our method. Implementing sparse convolution with actual inference speed improvement is not easy due to memory access overhead [9,10,56–58], while DeltaCNN [11] achieves actual inference speedups on GPUs through an elaborate CUDA implementation. MotionDeltaCNN [12] further improves the efficiency of DeltaCNN by using correspondence points under conditions of known camera motion. Unlike these methods, our approach is built on ViT, which has recently demonstrated effectiveness across a variety of computer vision tasks. Our method provides actual speedup on both CPUs and GPUs without requiring complex implementation, thanks to the nature of ViT, where the simple reduction of tokens directly reduces the actual computational cost.

Although there are recent studies that conduct token reduction in ViT to improve the efficiency of video processing, most of them [59–62] are designed for offline video processing, where all frames, including future ones, are accessible. This is different from online video processing, which we focus on. A concurrent study [63] conducts token reduction for online video recognition, but it is limited to the object tracking task, as tokens are selected based on past detection results. In contrast, our method is task-agnostic and can be used as the backbone for a wide range of online video understanding tasks due to its simplicity.

3. Method

We propose RetinaViT as an efficient frame-level visual feature extractor for online video. RetinaViT is composed of Retina blocks that approximate the original Transformer block to process only tokens that have changed over time, called event tokens. RetinaViT has several approximation parameters that adjust computational costs. During training, we randomly vary the approximation parameters and obtain a single model that can extract the appropriate features with any setting of approximation parameters. This training procedure is also expected to have a regularization effect, as in previous work for image recognition [64]. The approximation parameters of the trained RetinaViT are then tuned using black-box optimization to maximize accuracy and computational efficiency.

In the following, we will first describe the approximation of the Transformer block (Section 3.1), and then the model training and tuning of the approximation parameters (Section 3.3).

3.1. Approximation of Transformer Block

Transformer block basics. ViT [13] divides an image into non-overlapping patches, encodes them as token embeddings via linear projection and positional embedding (i.e., patch encoding; see Appendix A), and analyzes them with Transformer blocks [14]. Let us denote token embeddings as $\boldsymbol{z}_l \in \mathbb{R}^{N \times D}$, where N is the number of tokens, D is the number of channels, and $l \in \{0, 1, \ldots, L\}$ is the depth in ViT, which consists of L Transformer blocks. Here, ($\boldsymbol{z}_0$ is the output from the initial patch encoding, while $\boldsymbol{z}_{l>0}$ is the output of the l-th Transformer block). In the l-th Transformer block, the token sequence is processed as follows:

$$\boldsymbol{q}_{l+1} = \boldsymbol{k}_{l+1} = \boldsymbol{v}_{l+1} = \mathrm{LN}(\boldsymbol{z}_l), \tag{1}$$

$$\boldsymbol{h}_{l+1} = \mathrm{MHA}(\boldsymbol{q}_{l+1}, \boldsymbol{k}_{l+1}, \boldsymbol{v}_{l+1}), \tag{2}$$

$$\boldsymbol{z}_{l+1} = \mathrm{MLP}(\mathrm{LN}(\boldsymbol{h}_{l+1})) + \boldsymbol{h}_{l+1}, \tag{3}$$

where LN is layer normalization [65] (Appendix B), MHA is multi-head attention [14] (Appendix C), and MLP is a two-layer token-level MLP, including the GELU activation function [66] with a hidden size of $4D$. $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$ correspond to the queries, keys, and values that are inputs to multi-head attention, respectively (Appendix C). The original Transformer block uses what is known as self-multi-head attention, where queries, keys, and values are all identical, as shown in Equation (1) [13]. We refer to keys and values as "contexts".

Retina block. In videos, pixel values are often very similar within the spatiotemporal neighborhood. The same tendency holds for the feature space, where features for the same spatial location are often similar. We introduce a method for the efficient approximation of the Transformer block by processing only tokens with significant changes from the previous frame—event tokens—and skip the processing of the remaining tokens, which we call non-event tokens.

We refer to this approximated Transformer block as "RetinaBlock" and provide an overview of its process in Figure 2. First, we introduce the "event score", which is the L2 distance between a token in the current frame and its spatial counterpart in the previous frame, and determine event tokens by thresholding this score. The indices of event tokens in the l-th token sequence $\boldsymbol{z}_l^t$ at time t are determined as follows:

$$S_l^t = \left\{ i \in \{1, \ldots, N\} \mid i \leq N_\mathrm{s} \vee d_{t,(i)}^l < \delta_l \right\}, \tag{4}$$

$$d_{l,(i)}^t = ||\boldsymbol{z}_{l,(i)}^t - \boldsymbol{z}_{l,(i)}^{t-1}||_2, \tag{5}$$

where $\boldsymbol{z}_{l,(i)}^t$ is the i-th token in the token sequence $\boldsymbol{z}_l^t$, $d_{l,(i)}^t$ is the event score, and N_s is the number of special tokens (e.g., classification token [13]) that are not derived from the input image patch. These tokens are always treated as event tokens.

In the Retina block, Equations (1)–(3) are modified as follows:

$$\boldsymbol{q}_{l+1} = \mathrm{LN}(\boldsymbol{z}^t_{l,i\in S^t_l}),\ \boldsymbol{k}_{l+1} = \boldsymbol{v}_{l+1} = \mathrm{LN}(\boldsymbol{z}^t_l), \tag{6}$$

$$\boldsymbol{h}_{l+1} = \mathrm{MHA}(\boldsymbol{q}_{l+1}, \boldsymbol{k}_{l+1}, \boldsymbol{v}_{l+1}), \tag{7}$$

$$\boldsymbol{z}^t_{l+1,i\in S^t_l} = \mathrm{MLP}(\mathrm{LN}(\boldsymbol{h}_{l+1})) + \boldsymbol{h}_{l+1}, \tag{8}$$

$$\boldsymbol{z}^t_{l+1} = \mathrm{Align}\left[\boldsymbol{z}^t_{l+1,i\in S^t_l}, \boldsymbol{z}^{t-1}_{l+1,i\in \overline{S^t_l}}\right], \tag{9}$$

where Align$[\cdot]$ denotes the concatenation of token sequences and their alignment to the original token order. Multi-head attention only takes event tokens as queries and all tokens as contexts (Equation (7)), followed by a token-level MLP (Equation (8)).

In multi-head attention, each query token is processed independently of other query tokens, relying solely on itself and the contexts. Similarly, the MLP processes each token independently. Therefore, the outputs for event tokens are identical to those processed by the vanilla Transformer block. By combining the event tokens with the non-event tokens, which are carried over from the previous timestep, we obtain the final output (Equation (9)).

Since the theoretical computational complexity of the Transformer block is (roughly) proportional to the number of query tokens [64,67], the computational cost can be reduced by increasing the number of non-event tokens.

We construct RetinaViT by stacking multiple Retina blocks and train it end-to-end. Once a token becomes a non-event token in RetinaViT, it never reverts to an event token because its difference with the corresponding token at the previous time is zero (as it is replaced by itself). On the other hand, even if a token is an event token in an earlier block, as the feature extraction progresses, the event score decreases, and the token may become a non-event token. This can be seen as an adaptive adjustment of the number of blocks to process for each token.

Note that at the first timestep $t = 0$, any tokens are not dropped, i.e., all tokens are processed, and their information is reused in subsequent timesteps.

Local context tokens. In frame images, spatially close areas are more strongly related than distant areas. Following this characteristic, there are methods to improve the efficiency of Vision Transformers (ViTs) by limiting attention to spatially local areas [18,68]. Similarly, we consider restricting attention to the spatial locality around each event token to achieve a more aggressive reduction in computational costs. However, existing local attention methods are designed for densely arranged tokens and cannot effectively improve inference speed for sparse tokens due to memory access costs. Implementing efficient local attention for sparsely arranged event tokens is not trivial.

We introduce local attention to the Retina block with a simple yet practically effective implementation. Specifically, we consider the union set of spatial neighborhoods of event tokens as the context tokens (see Figure 2). Equation (1) is, thus, modified as follows:

$$\boldsymbol{k}^t_{l+1} = \boldsymbol{v}^t_{l+1} = \mathrm{LN}(\boldsymbol{z}^t_{l,\phi_{m_l}(S^l_t)}), \tag{10}$$

where ϕ_{m_l} is a function that extends the index to a spatial neighborhood, and m_l is the number of neighbors in the l-th block.

In existing local attention methods [18,68], only the neighboring tokens of each token are used as context tokens. In contrast, our method includes the neighboring tokens of other event tokens as well, thus broadening the context for each event token. This approach leaves room for theoretical reductions in computational costs. Although memory-access overhead is minimized and the number of contexts correlates with theoretical computational costs, this implementation is effective enough to practically improve computational efficiency.

3.2. RetinaViT

RetinaViT is a stack of multiple Retina blocks as shown in Figure 1. We construct multiple variants of RetinaViT by varying the size of the Transformer block in the Retina

block while keeping the patch size at 16×16, as summarized in Table 1. Following the original naming convention, we refer to RetinaViT variants using the name of the architecture and the patch size; for example, RetinaViT-B16 refers to the "Base" model with a patch size of 16×16. Each variant of RetinaViT has the same Transformer block configuration as the corresponding original ViT (e.g., RetinaViT-B16 corresponds to ViT-B16); therefore, it is possible to transfer the weights of the corresponding pre-trained ViT to RetinaViT.

Table 1. RetinaViT variants. The number of blocks (L) and the configuration of the Transformer block (the number of channels D and the number of heads H) are shown.

Model Name	# Blocks (L)	# Channels (D)	# Heads (H)	# Params
Tiny	12	192	3	5.7 M
Small	12	384	6	22.1 M
Base	12	768	12	86.6 M

For downstream tasks, we can extract frame-level features in multiple ways (all tokens or classification tokens) at multiple stages (final layer or intermediate layers) from RetinaViT. In the experiments, we selected the appropriate method for each downstream task, following the standard practice of the task and the decoder used. Details are described in Section 4.1.

3.3. Model Training and Approximation Parameter Optimization

RetinaViT has approximation parameters to adjust computational costs, including thresholds $\{\delta_l\}_{l=1}^{L}$ for the event score and the neighborhood numbers of context tokens $\{m_l\}_{l=1}^{L}$. We determine these approximation parameters by a train-then-optimize strategy; we first train RetinaViT so that it can extract appropriate features with any approximation parameter setting, and then fix the trained model and optimize the approximation parameters using black-box optimization to maximize both accuracy and efficiency. The details of training and optimization are as follows.

Model training. We randomly vary the approximation parameters $\theta = \{\{\delta_l\}_{l=1}^{L}, \{m_l\}_{l=1}^{L}\}$ over a defined range during training, obtaining a model that can extract features appropriately with any approximation parameter setting in that range. m_l varied as an integer in the range $[0, 4]$. δ_l depends on the distribution of the event score, which varies during training. Therefore, we pick up tokens with event scores in the r percentile as event tokens. We set r in the range $[0, 1]$.

In this way, the model undergoes various approximation parameter settings during training and can properly infer under any of these settings. Our training method is the same as standard training except for the random variation of these parameters and does not require any additional losses other than the main loss. As in previous work for image recognition [64], this training procedure is also expected to have a regularization effect.

Approximation parameters optimization. After model training, we obtain the approximation parameters θ of the trained model $F_\theta(\cdot)$ by maximizing accuracy and computational efficiency using black-box optimization. We solve the following multiobjective optimization:

$$\begin{aligned} &\underset{\theta}{\text{maximize}} \quad \big(P,\ -C\big), \\ &P = \mathcal{P}(F_\theta(\cdot), \mathcal{D}_{\text{train}}),\ C = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}\left[\mathcal{C}(F_\theta(\cdot), x)\right], \end{aligned} \tag{11}$$

where $\mathcal{D}_{\text{train}}$ is the training data, x is the input and y is the target in the training data, respectively. $\mathcal{P}(F_\theta(\cdot), \mathcal{D}_{\text{train}})$ is a task-specific evaluation metric when applying $F_\theta(\cdot)$ to $\mathcal{D}_{\text{train}}$ (e.g., frame-level accuracy for online action recognition, IoU for video object segmentation), and $\mathcal{C}(F_\theta(\cdot), x)$ is the computational cost (e.g., FLOPs, runtime) when x is processed by $F_\theta(\cdot)$. In real application cases, the optimization problem can be modified according to the scenario's requirements; for example, one can formulate a constrained optimization that

maximizes accuracy by setting an upper bound on computational cost. Notably, in our approach, once model training—the most time-consuming process—is complete, we can obtain models with different computational costs by changing the constraints and optimizing the approximation parameters. In the experiments, we solve the above multiobjective optimization using MOTPE [69] and draw the Pareto front between accuracy and efficiency.

4. Experiments

We conducted extensive evaluations on three online video understanding tasks to assess the effectiveness of RetinaViT. The rest of this section is organized as follows: first, we describe the setup for the three evaluation tasks in Section 4.1, followed by the task-agnostic implementation details in Section 4.2. Next, we provide the evaluation results for each component of RetinaViT, the analysis of the training procedure, and a comparative study with existing methods in Sections 4.3, 4.4, and 4.5, respectively. Finally, we provide visualizations of event scores, event tokens, and predictions to discuss the behavior of RetinaViT in Section 4.6.

4.1. Task Setups

RetinaViT can be applied as a frame-level visual feature encoder for a variety of online video understanding tasks. To demonstrate its effectiveness and generalizability, we evaluate RetinaViT on the following three tasks, each with different characteristics.

Online action recognition is the task of classifying the classes of actions taking place at each frame in an online video. We used the 50Salads dataset [70], which contains 50 videos in which 25 people prepare two kinds of mixed salads. The videos, each consisting of 9000 to 18,000 RGB frames, are annotated with 17 action classes per frame. We built RetinaViT-T16/S16/B16 based on the original ViTs pre-trained on ImageNet-21k [71,72]. Each frame was processed independently in a sliding-window manner, and we used the classification token as the frame-level feature. The final classification result for each frame was obtained by inputting the frame-level feature into a linear decoder. Although we could use more sophisticated decoders (e.g., LSTM, Transformer) that take temporal information into account, we used the simplest one to measure the pure performance of feature encoders. We used cross-entropy as the loss function. Following previous works [73], we downsampled the videos in the 50Salads dataset from 30 fps to 15 fps. The video was divided into non-overlapping chunks of 20 frames. For training, the frames were cropped randomly but consistently throughout each chunk and resized to 224×224. For testing, the frames were center-cropped and resized to 224×224. We set the maximum number of epochs to 25 and the batch size to 8. We used Top-1 accuracy as the evaluation metric.

Online pose estimation is the task of predicting the key point coordinates of a person in each frame of an online video. We evaluated Sub-JHMDB [74], which is a collection of 11,200 frames from 316 video clips, labeled with 15 body joints.

The video was divided into non-overlapping chunks of 20 frames. As in previous studies [8,10,25,26], we cropped the frame images based on the bounding box of the person obtained from the puppet masks in the dataset. The cropped images were then resized to 256×192 for model input. A random flip was performed for data augmentation during training. Models were required to output the keypoint coordinates of the person from these cropped frame images. We set the maximum number of epochs to 200 and the batch size to 4. We adopted the PCK introduced in [75] as a metric. Each predicted coordinate was considered correct if it was within $\alpha \cdot \max(h, w)$ from the true position, where h and w are the height and width of the bounding box. We set α to 0.2.

We built RetinaViT-S16/B16 based on ViTPose [15] MS COCO [76]. In ViTPose, the original ViTs were used as the backbone; therefore, we could transfer the weights of the ViT part to RetinaViT. Following the original ViTPose, we used the output of the last block as the frame-level feature and upsampled it with the decoder consisting of three deconvolution layers to output keypoint coordinates in heatmap format. We used the mean squared error on heatmaps as the loss function.

Video object segmentation is a task in which the model outputs the segmentation mask of the object in each frame, given the segmentation mask for the first frame in an online video. For evaluation, we used DAVIS 2017 [77], which contains 60 training videos, 30 validation videos, and 30 test-dev videos. We constructed the methods based on BMVOS [6], which consists of multi-scale frame-level feature extraction using DenseNet and a decoder that considers the relationship between frame features. We conducted the evaluation by replacing the DenseNet encoder with RetinaViT-T16/S16, which were based on the pre-trained ViTs on ImageNet-21k as in the action recognition task. For multi-scale frame-level feature extraction, we upsampled the output of the 4th, 8th, and 12th blocks to different resolutions to fit the BMVOS decoder with deconvolutions, as conducted in DPT [16]. We set the maximum number of epochs to 100 and the batch size to 8. We cropped the region randomly but consistently throughout each video sequence and resized it to 384×384 during training, and resized the frame to a height of 384 while maintaining the aspect ratio during testing.

Following the previous work [6], we used the $\mathcal{G}$ score, which is the average of the segment intersection-over-union (IoU) and the boundary IoU, for evaluation [77].

4.2. Implementation Details

We used Adam [78] for training on all three tasks with a learning rate of 10^{-4}. To tune the approximation parameters, we used the metric function of each task as the performance metric $\mathcal{J}(\cdot)$ and GPU inference time as the computational cost $\mathcal{B}(\cdot)$. Although FLOPs can be used as a measure of computational cost, optimizing for actual inference time produced a better trade-off, as the actual inference time has some overhead that cannot be measured by FLOPs. In real-world applications, it should be possible to measure and optimize inference time for each device actually used to achieve a better trade-off. While our method also supports constrained optimization of computational cost, which is thought to be useful in real applications, we solved the multiobjective optimization in Equation (11) to obtain the Pareto front using MOTPE [69] for a detailed evaluation of the trade-off between accuracy and efficiency. Specifically, we first obtained the Pareto front on the training split by optimization with 30 trials, then drew the Pareto front on the test (or validation) split by evaluating the test (or validation) split using the approximation parameters of each point of the Pareto front on the training split. Note that in this phase, the model was in inference mode, so one trial was faster than one epoch in model training. Unless otherwise specified, we applied event token detection at the first and fourth blocks, as determined by the ablation study (Section 4.3).

The actual inference time was measured on both a CPU and a single GPU. We used an Intel(R) Core(TM) i9-10980XE CPU @ 3.00 GHz (Intel Corporation, Santa Clara, CA, USA) for the CPU and an NVIDIA RTX A5000 (NVIDIA Corporation, Santa Clara, CA, USA) for the GPU. For inference on the CPU, we set the batch size to 1. For the GPU, a small batch size did not make a significant difference in inference time because the processing power of the device could not be fully utilized (the same was reported in [11]), so we set the batch size to 64 and reported the average per sample inference time.

4.3. Ablation Study

Sanity check. To confirm the basic validity of the proposed method, we compare it with systematically constructed baselines on the action recognition task. Specifically, we consider multiple patterns in the training and inference phases. First, for training, in addition to our (1) event-based approach, where tokens are dropped based on event score and randomly sampled thresholds, the following can be considered: (2) vanilla: the training strategy used in the original ViT (i.e., tokens are not dropped), and (3) random: a training method that ignores event score and drops tokens completely at random. For inference, we consider our (1) event-based approach, where event tokens are selected by thresholding the event score, and (2) random, where event tokens are selected completely at random.

The results are shown in Figure 3. First, we can see that our method achieves a significant speed increase over the original ViT with accuracy improvements (rather than degradations). We speculate that this is due to the improved robustness of focusing only on the changed areas and the regularization effect of randomly dropping tokens, as in [64]. Second, when training with the "vanilla" approach, we observe an accuracy drop as the number of tokens decreases because the model is not trained to handle token drops. On the other hand, the model trained with the "random" strategy assumes token drops, leading to some improvement in the trade-off during "event-based" inference, but it is not as effective as the proposed method. These comparisons indicate that the event score is meaningful for determining the importance of a token.

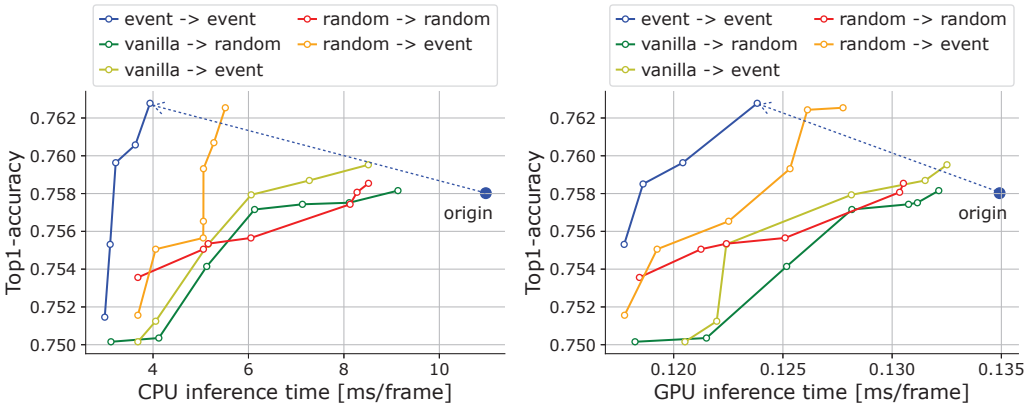


Figure 3. Sanity check on 50Salads [70]. We present trade-off curves between the accuracy and inference time on the CPU (**left**) and GPU (**right**). In the legends, the start and end points of each arrow represent training and inference strategies, respectively. "Origin" in each graph represents the result where no token selection is used (original ViT). We draw dashed arrows representing the improvements in the trade-off between our method (event-based -> event-based) and the corresponding "origin".

Locations of event token detection. The detection of event tokens can be conducted in any block of RetinaViT. However, if event token detection is performed in too many blocks, the number of event tokens may become extremely small during training, which could be problematic. Therefore, we compared the cases in which event tokens are detected in only the first block, the first and fourth blocks, and all blocks. As can be seen from Figure 4, the accuracy deteriorates when event token detection is performed in all blocks. We believe this is due to the extremely small number of tokens that the model can consider (i.e., event tokens) during training, as noted above. Although the case involving the first block alone recorded a decent trade-off, the case involving the first and fourth blocks, in which event tokens could be determined at multiple phases, was found to have the best trade-off.

Local context tokens. We compare RetinaViT with and without local context tokens in Figure 5. We can see that the trade-off is improved by introducing local context tokens. The introduction of locality in ViT, as per existing studies [18,68], is found to be effective in terms of increasing efficiency.

Fine-tuning after approximation parameters optimization. We have the option of further fine-tuning after optimizing approximation parameters, as is conducted in pruning methods [40,79], namely post-fine-tuning. The evaluation results of this option are shown in Figure 5. While post-fine-tuning improves accuracy, the gain is not large. The model trained by our method has already undergone all possible approximation parameter settings and is, therefore, ready to work without post-fine-tuning. Since post-fine-tuning for each approximation parameter setting is labor- and computation-intensive, we did not use it in the following experiments.

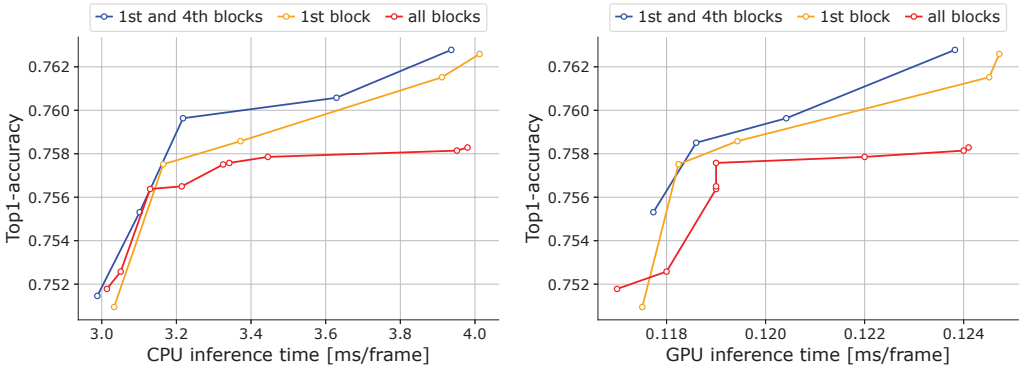


Figure 4. Ablation results for the locations of event token detection on 50Salads [70]. We show trade-off curves between the accuracy and inference time on the CPU (**left**) and GPU (**right**). Note that we do not show "origin" (original ViT) in this figure to clarify the differences, but we have successfully improved the trade-off significantly compared to the "origin" as shown in Figure 3.

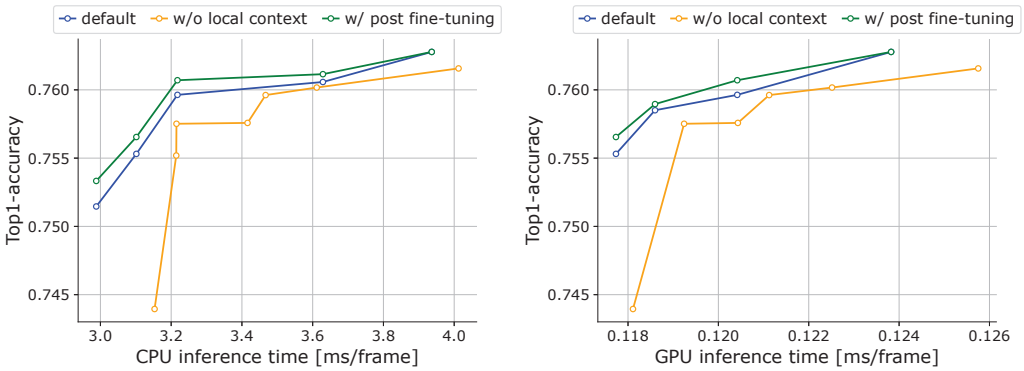


Figure 5. Ablation results for local context tokens and post-fine-tuning on 50Salads [70]. We show trade-off curves between the accuracy and inference time on the CPU (**left**) and GPU (**right**). Note that we do not show "origin" (original ViT) in this figure to clarify the differences, but we have successfully improved the trade-off significantly compared to the "origin" as shown in Figure 3.

4.4. Analysis of the Training Procedure

The training of RetinaViT can be considered as a form of random information dropping, which may have a regularization effect, as reported in previous studies [64,80,81]. To investigate this, we compared the learning curves of RetinaViT with those of the corresponding original ViT in Figure 6. We can see that the learning curves on the validation sets of RetinaViT are relatively stable across all datasets. This regularization effect is considered to be one of the reasons for the slight performance improvement over the original ViTs (Figure 3).

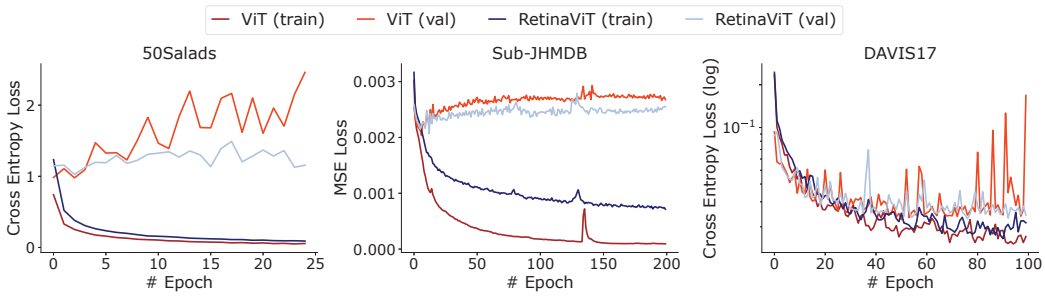


Figure 6. Comparisons of the learning curves between RetinaViT-S16 and the original ViT-S16. The vertical axis represents the loss for each task, and the horizontal axis represents the number of epochs. In all datasets, the validation learning curves of RetinaViT are relatively stable.

4.5. Comparative Study with Existing Methods

Online action recognition. We implemented the widely used visual backbones ResNet-50/101/152 [82], Swin Transformer-T16/S16/B16 [18], ViT-T16/S16/B16 [13], and RetinaViT-T16/S16/B16 for comparison, and ResNets are pre-trained on ImageNet and the others are pre-trained on ImageNet-21k. All models are used as frame-level classifiers without any special decoder, such as LSTM. We also applied DeltaCNN [11], a method for CNNs that processes only those parts of the frame that have changed, similar to the proposed method, to ResNets. Since there is only a CUDA implementation of DeltaCNN, we evaluated it only on a GPU. DeltaCNN, like our method, has thresholds that adjust the computational costs, and we determined them using the same black-box optimization procedure as RetinaViT to ensure a fair comparison. The comparisons are shown in Figure 7. First, compared to original ViTs, RetinaViT achieved significant improvements in efficiency with better accuracy and the improvement in efficiency is more pronounced when the model is larger. In particular, when we used ViT-B16 as a base model, RetinaViT achieved 61.9% and 50.8% reductions in inference time with slight accuracy improvements rather than degradation in action recognition. For ViT-T16, which is the smallest model, the efficiency improvement on the GPU is smaller. We attribute this to the fact that the GPU's computational capacity is sufficiently large compared to the model's computational cost, making the difference insignificant. As noted in Section 4.3, we attribute the improvement in accuracy to the improved robustness by focusing on changed locations and the regularization effect of random token dropping. Due to the strong representational capabilities of the base models, ViTs, RetinaViT offers a better trade-off compared to ResNets and ResNets equipped with DeltaCNN. While Swin Transformers show strong trade-offs, RetinaViT achieves comparative accuracy even in the low-computational-cost range (0.25 ms/frame in GPU inference time, 20 ms/frame in CPU inference time), where the smallest model of Swin Transformers, T16, cannot be reached. We believe that RetinaViT can be the best option for applications that require such a low computational cost range.

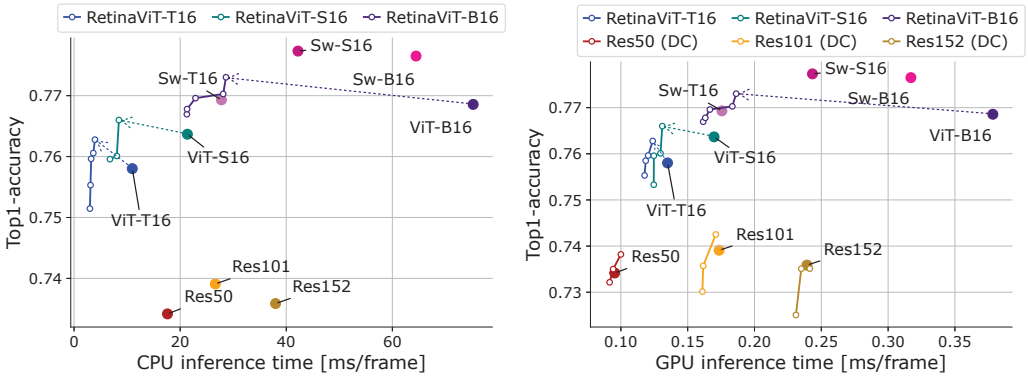


Figure 7. Comparisons of the trade-off between the accuracy and inference time on 50Salads [70] (val). Inference time was measured on the CPU (**left**) and GPU (**right**). "Sw" and "Res" represent Swin Transformer [18] and ResNet [82], respectively. "DC" represents DeltaCNN [11], which we only use on the GPU as it does not support CPU inference. Dashed arrows represent the improvements in the trade-off between RetinaViT and the corresponding original ViT.

Online pose estimation. We compare our RetinaViT with ViTPose-S16/B16 (which are the base models of RetinaViT) and HR-Nets (w32, w48) [83]. We also introduce DeltaCNN to HR-Nets, as we did with the action recognition evaluation. All models were pre-trained on MS COCO [76]. We show the results in Figure 8. RetinaViT achieved higher efficiency with equal or better accuracy than their respective base models (ViTPose), especially on the CPU, and demonstrated a superior trade-off compared to HR-Nets and HR-Nets equipped with DeltaCNN. Compared to the case of action recognition, the improvements from the base models were not large. This is likely because dropping tokens more strongly affects

accuracy due to the nature of pose estimation, where fine-grained features are more crucial to predict keypoint coordinates, which we also discuss in Section 4.6.

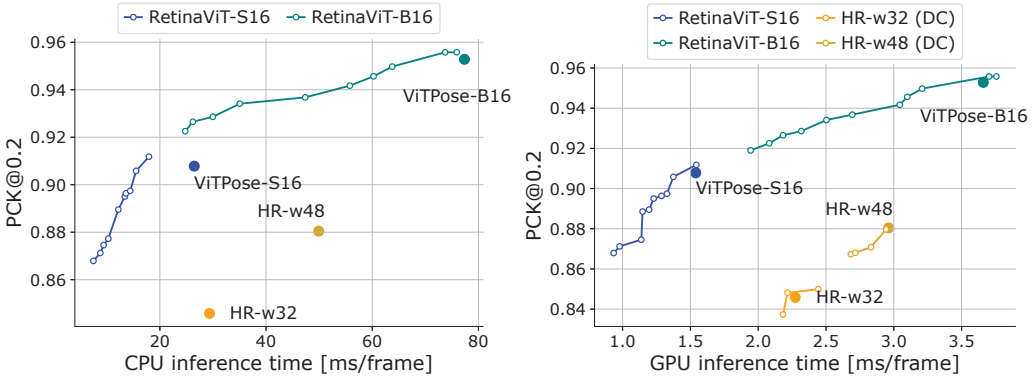


Figure 8. Comparisons of the trade-off between accuracy (PCK@0.2 [75]) and inference time on Sub-JHMDB [74] (val). Inference time was measured on the CPU (**left**) and GPU (**right**). "HR" represents HR-Net [83]. "DC" represents DeltaCNN [11], which we only use on the GPU as it does not support CPU inference.

Video object segmentation. Finally, we evaluated VOS, replacing the frame encoder of BMVOS. The detailed settings are described in Section 4.1. We show the results in Figure 9. We can see that the model significantly improves computational efficiency without severe accuracy degradation, especially when using ViT-S16. Even though VOS requires more fine-grained features than pose estimation, RetinaViT shows a certain effectiveness. The reason why the computational efficiency does not improve much with ViT-B16 is that the model's computational cost is too small compared to the device's computational capacity, making the impact of token reduction on inference speed small, as mentioned in online action segmentation. Compared to BMVOS, RetinaViT shows a comparable trade-off. Given that ViTs scale better in performance with the training data numbers compared to CNNs, and that larger ViT-based pre-trained models are expected to emerge in future research, we would like to emphasize the importance of improving trade-offs in ViTs.

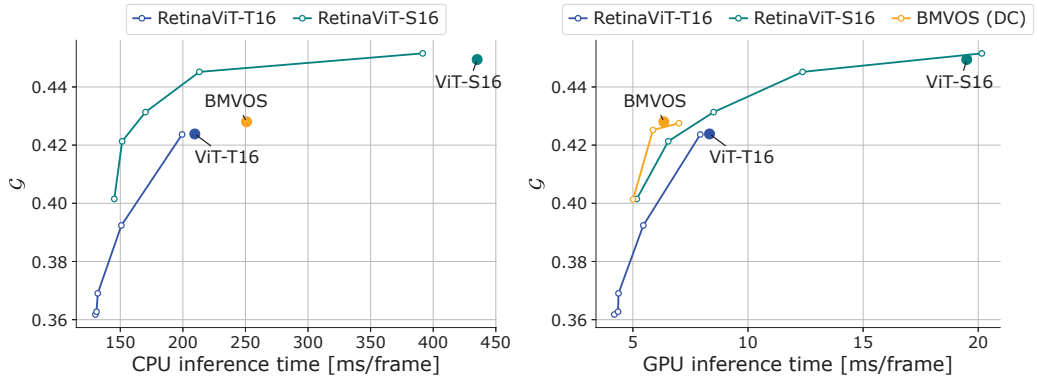


Figure 9. Trade-off comparisons between the accuracy ($\mathcal{G}$ score [77]) and inference time on DAVIS17 [77] (test-dev). Inference time was measured on the CPU (**left**) and GPU (**right**). "DC" represents DeltaCNN [11], which we only use on the GPU as it does not support CPU inference.

4.6. Visualization

We visualize event scores, event tokens, and predictions in videos from 50Salads [70] (Figure 10), Sub-JHMDB [74] (Figure 11), and DAVIS2017 [77] (Figure 12). In 50Salads, we observe that many tokens have low event scores, and dropping them does not degrade accuracy. For Sub-JHMDB, there are fewer tokens with significantly low event scores compared to 50Salads. This is primarily due to the large motions of the subjects and the

perturbation caused by frame-level cropping. Although there is no significant degradation in predictions within these samples, the more aggressive token reduction can degrade accuracy, limiting the reducible computational costs. For DAVIS2017, in the top two videos, many tokens have low event scores, allowing a large number of tokens to be reduced without degrading the predictions. However, in the bottom two videos, the event scores are generally high due to large camera motions. We visualize predictions with two different thresholds for these videos (see the "drop" in the last two videos in Figure 12). When attempting to reduce a large number of tokens (similar to the top two samples) in these videos, the target objects are also dropped, causing significant accuracy degradation (see the second "drop" in each video).

We are interested in exploring frame-to-frame correspondence to mitigate the effects of camera motions and the perturbation caused by cropping in future work. Additionally, instead of a hard decision (event token or not), a method that determines the weight of processing in a multi-step or continuous manner according to event scores has the potential to address "subtle" events caused by rapid camera motions and frame-level cropping.

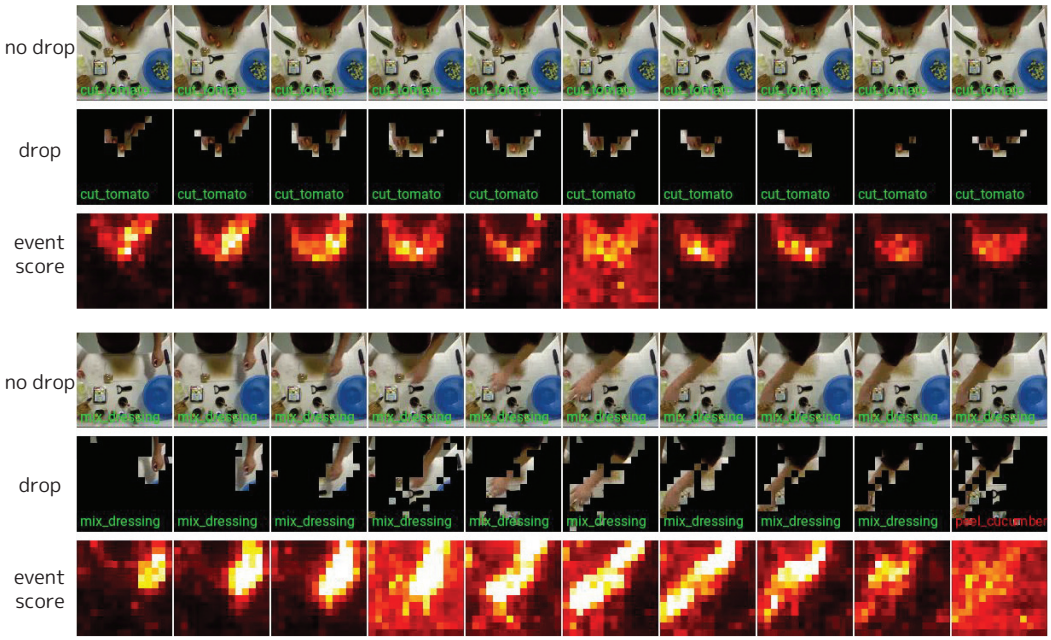


Figure 10. Visualization of event scores, event tokens, and predictions on 50Salads [70]. The frames are arranged from left to right in time order, and each column represents the same timestep. "no drop" represents the prediction without dropping tokens (i.e., $\delta_l = 0$ for all l) overlaid on the input frames. "drop" represents the event tokens in the fourth block, on which the corresponding prediction is overlaid. For visibility, we overlaid the locations of event tokens on the corresponding input RGB frames, where the non-event tokens are blacked out. The predictions are drawn in green if correct and in red if incorrect. "event score" represents the event scores as a heatmap at the input for the fourth block. Note that all tokens are processed in the first frame of each video clip, which is not depicted in the figure.

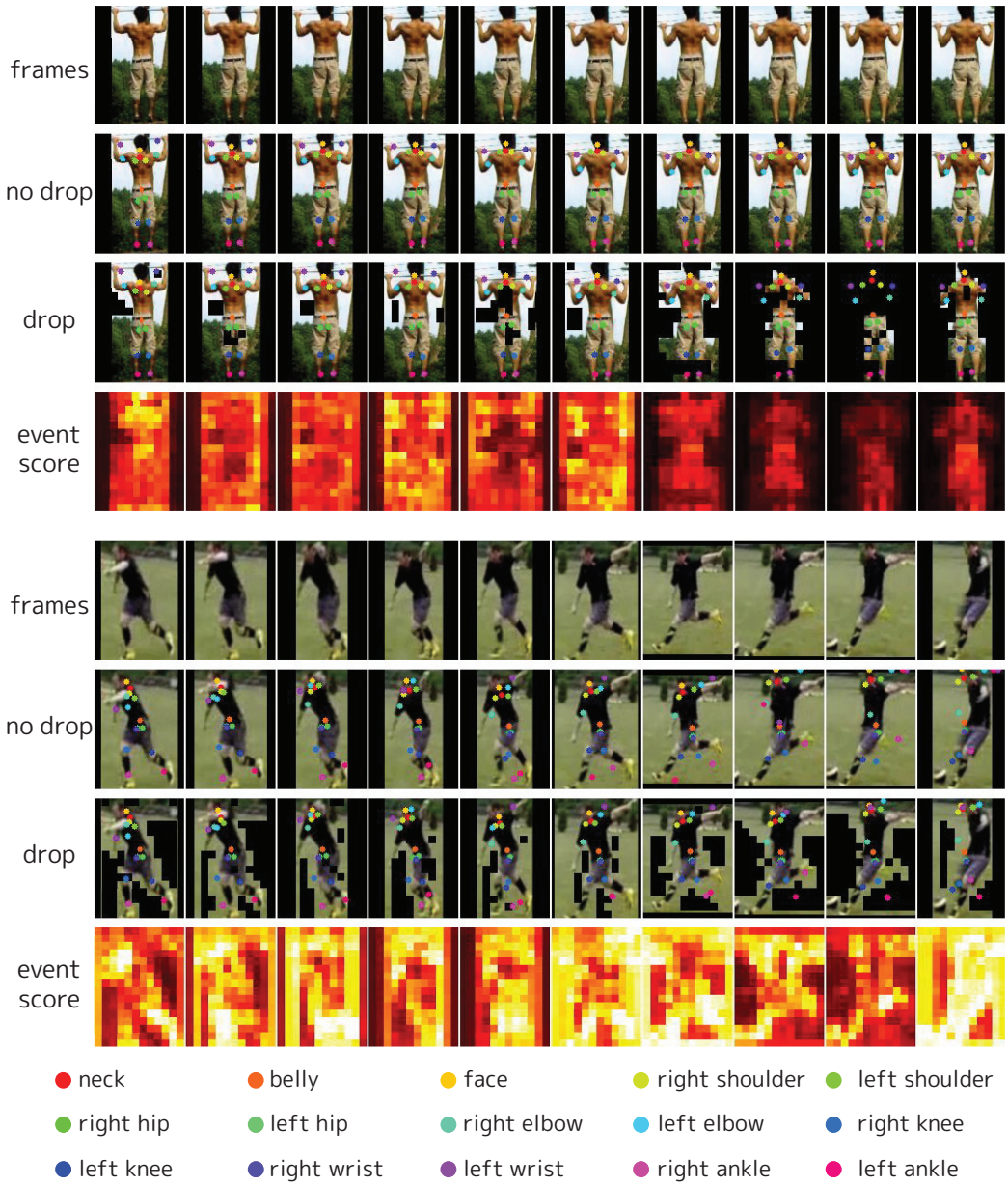


Figure 11. Visualization of event scores, event tokens, and predictions on Sub-JHMDB [84]. The frames are arranged from left to right in time order, and each column represents the same timestep. "no drop" represents the prediction (the locations of the key points) without dropping tokens (i.e., $\delta_l = 0$ for all l), overlaid on the input frames. "drop" represents the event tokens in the fourth block, on which the corresponding prediction is overlaid. For visibility, we overlaid the locations of event tokens on the corresponding input RGB frames, where the non-event tokens are blacked out. "event score" represents the event scores as a heatmap at the input for the fourth block. Note that all tokens are processed in the first frame of each video clip, which is not depicted in the figure.

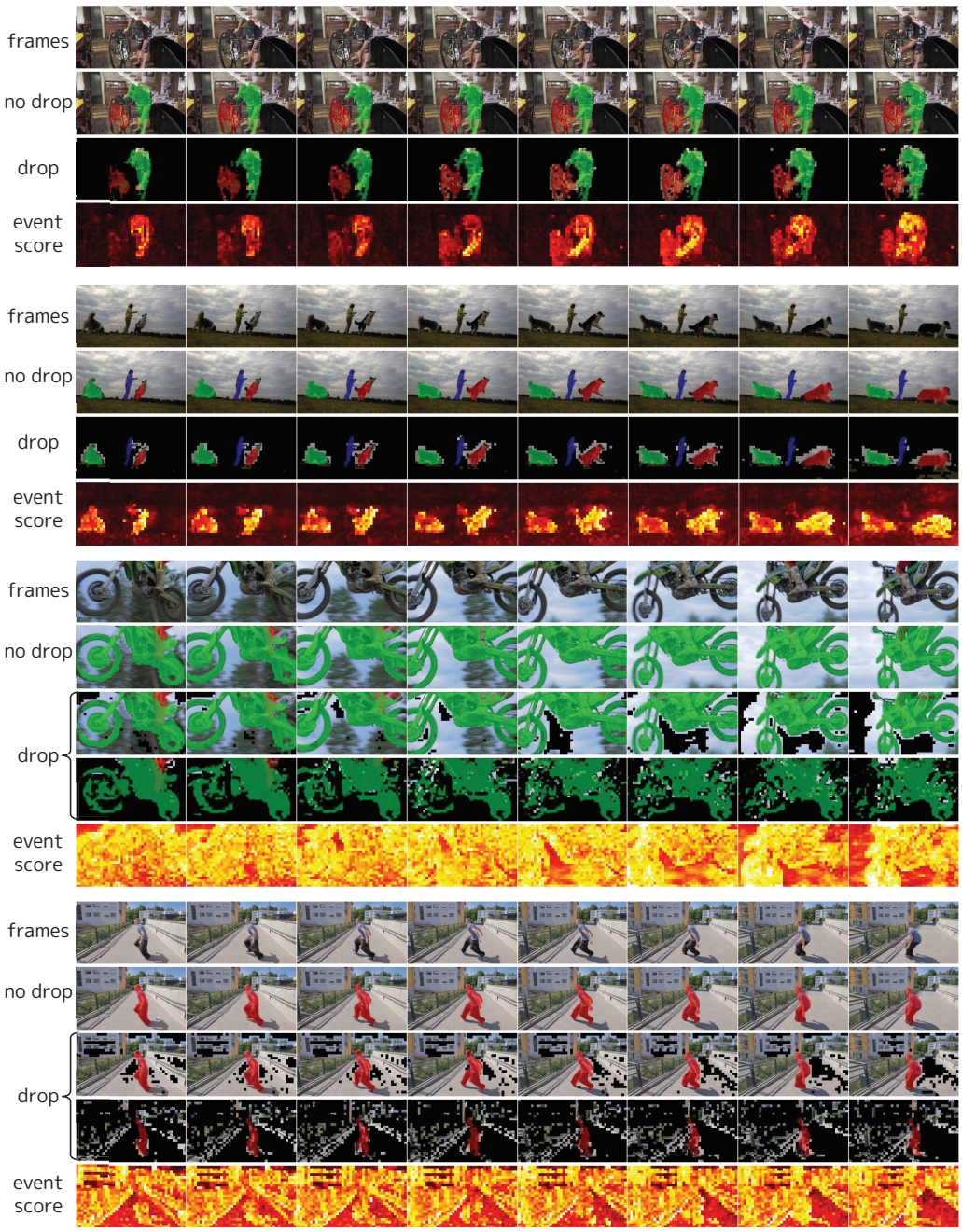


Figure 12. Visualization of event scores, event tokens, and predictions on DAVIS2017 [77]. The frames are arranged from left to right in time order, and each column represents the same timestep. "no drop" represents the prediction masks without dropping tokens (i.e., $\delta_l = 0$ for all l), overlaid on the input frames. "drop" represents the event tokens in the fourth block, on which the corresponding prediction is overlaid. For visibility, we overlaid the locations of event tokens on the corresponding input RGB frames, where the non-event tokens are blacked out. The prediction masks are drawn in different colors for each instance. "event score" represents the event scores as a heatmap at the input for the fourth block. For the last two videos, we show two "drops" with different thresholds δ_l as examples where it is difficult to reduce computational costs while maintaining accuracy due to large camera motion. Note that all tokens are processed in the first frame of each video clip, which is not depicted in the figure.

5. Conclusions

We propose RetinaViT, which efficiently extracts frame-level visual features in online video streams by exploiting redundancy in videos. RetinaViT is task-agnostic, leverages the strong representational power of ViTs, and achieves improvements in actual inference speed. We conducted extensive experiments on action recognition, pose estimation, and object segmentation in online videos, demonstrating that RetinaViT improves the trade-off between accuracy and efficiency.

In the future, for tasks involving fine-grained features, such as pose estimation and object segmentation, we are interested in investigating methods to exploit frame-to-frame correspondence to mitigate the effects of camera motions and the perturbation caused by cropping. Additionally, methods to complement excluded information (i.e., non-event tokens), such as merging them at the end of the model and using a soft definition of event tokens, might be effective. Although this study focused on a task-agnostic frame-level feature encoder, it is valuable to modify the decoder of each task to leverage event tokens for further improvement in computational efficiency. A meaningful direction will be to extend RetinaViT to ViT-based space-time visual backbones [35,36] for clip-level features.

Author Contributions: Conceptualization, T.S.; formal analysis, T.S.; investigation, T.S.; methodology, T.S.; resources, Y.A.; software, T.S.; supervision, Y.A.; validation, T.S.; visualization, T.S.; writing—original draft, T.S.; writing—review and editing, T.S. and Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are openly available in reference number [70–72,74,77]. The original contributions presented in this study are included in the article and further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
ViT	Vision Transformer
FLOPs	floating point operations
GPU	graphics processing unit
CPU	central processing unit
IoU	intersection-over-Union

Appendix A. Patch Encoding and Special Tokens in ViT

Before the Transformer blocks process, ViT [13] converts an input image into a token sequence, called patch encoding. Firstly, an input image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ is divided into patches and reshaped into $\boldsymbol{x}_\mathrm{p} \in \mathbb{R}^{N \times (P^2 C)}$, where H, W are the height and width of the image, C denotes the channel, P denotes the side length of each patch, and $N = HW/P^2$ denotes the resulting number of patches. Subsequently, each patch is mapped to a D dimensional feature via a token-level linear projection $\phi : \mathbb{R}^C \mapsto \mathbb{R}^D$ and the corresponding positional embeddings are added to it:

$$\boldsymbol{z}_{\mathrm{p},(i)} = \phi(\boldsymbol{x}_{\mathrm{p},(i)}) + \boldsymbol{e}_{\mathrm{pos},(i)}, \tag{A1}$$

where i is an index of the token, $\boldsymbol{x}_{\mathrm{p},(i)}$ is the i-th element of $\boldsymbol{x}_\mathrm{p}$, and $\boldsymbol{e}_{\mathrm{pos},(i)}$ is the i-th element of the positional embeddings $\boldsymbol{e}_\mathrm{pos} \in \mathbb{R}^{N \times D}$.

Finally, we concatenate $\{z_{\mathrm{p},(i)}\}_{i=1}^{N}$ and special tokens at the token axis to obtain the input to the first Transformer block. In the original ViT, which we used as our base model, a trainable classification token $z_{\mathrm{cls}} \in \mathbb{R}^{1\times D}$ is added as a special token, as follows:

$$z_0 = \mathrm{Concat}\left[z_{\mathrm{cls}}, z_{\mathrm{p},(1)}, z_{\mathrm{p},(2)}, ..., z_{\mathrm{p},(N)}\right]. \tag{A2}$$

Appendix B. Layer Normalization

We use layer normalization (LN) [65] as in the original Transformer [14]. As stated in [81], LN in the Transformer is the same as channel normalization [85], and we follow it. Given an input tensor $\boldsymbol{x} \in \mathbb{R}^{N\times D}$ (N is the number of tokens and D is the number of channels), LN normalizes the tensor over the channel axis and scales and shifts it by trainable parameters:

$$\boldsymbol{y}_{i,j} = \frac{\boldsymbol{x}_{i,j} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \times \boldsymbol{\gamma}_j + \boldsymbol{\beta}_j, \tag{A3}$$

where $\mu \in \mathbb{R}^N$ and $\sigma \in \mathbb{R}^N$ are the mean and standard deviations of $\boldsymbol{x}$ over the channel axis, respectively, and $\gamma \in \mathbb{R}^D$ and $\boldsymbol{\beta} \in \mathbb{R}^D$ are trainable scaling and shifting parameters, respectively. ϵ is a small constant used to avoid division by zero and we set $\epsilon = 10^{-6}$.

Appendix C. Multi-Head Attention

We present the details of multi-head attention $\mathrm{MHA}(\cdot)$ [14]. Multi-head attention is composed of scaled dot-product attention (SDA) [14], which takes the query $Q \in \mathbb{R}^{N_\mathrm{q}\times D_\mathrm{q}}$, key $K \in \mathbb{R}^{N_\mathrm{kv}\times D_\mathrm{k}}$, and value $V \in \mathbb{R}^{N_\mathrm{kv}\times D_\mathrm{v}}$ as the input and sum V over the element axis, weighted by the attention computed from the query and key:

$$\mathrm{SDA}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{D_\mathrm{k}}})V. \tag{A4}$$

Let us denote $\boldsymbol{q} \in \mathbb{R}^{N\times D}$, $\boldsymbol{k} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{v} \in \mathbb{R}^{N\times D}$ as token sequences corresponding to the query, key, and value, respectively. In multi-head attention, they are linearly projected by H sets of trainable weights and independently fed into SDAs, and the H outputs are concatenated and linearly projected:

$$\mathrm{MHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = \mathrm{Concat}[o^1, o^2, ..., o^H]W_\mathrm{MHA}, \tag{A5}$$

$$o^h = \mathrm{SDA}(\boldsymbol{q}W_\mathrm{q}^h, \boldsymbol{k}W_\mathrm{k}^h, \boldsymbol{v}W_\mathrm{v}^h), \tag{A6}$$

where $W_\mathrm{q}^h \in \mathbb{R}^{D\times D_\mathrm{head}}$, $W_\mathrm{k}^h \in \mathbb{R}^{D\times D_\mathrm{head}}$, $W_\mathrm{v}^h \in \mathbb{R}^{D\times D_\mathrm{head}}$, and $W_\mathrm{MHA} \in \mathbb{R}^{H\cdot D_\mathrm{head}\times D}$ are trainable 2D matrices for the linear projections. D_head is typically set to D/H.

References

1. Hubel, D.H.; Wiesel, T.N. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **1959**, *148*, 574–591.
2. Posch, C.; Serrano-Gotarredona, T.; Linares-Barranco, B.; Delbrück, T. Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output. *Proc. IEEE* **2014**, *102*, 1470–1484. [CrossRef]
3. Sikora, T. The MPEG-4 video standard verification model. *IEEE Trans. Circuits Syst. Video Technol.* **1997**, *7*, 19–31. [CrossRef]
4. Zolfaghari, M.; Singh, K.; Brox, T. ECO: Efficient convolutional network for online video understanding. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
5. Xiao, B.; Wu, H.; Wei, Y. Simple Baselines for Human Pose Estimation and Tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
6. Cho, S.; Lee, H.; Kim, M.; Jang, S.; Lee, S. Pixel-Level Bijective Matching for Video Object Segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022.
7. Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; Lin, D. Temporal Action Detection with Structured Segment Networks. *Int. J. Comput. Vis.* **2020**, *128*, 74–95. [CrossRef]
8. Luo, Y.; Ren, J.S.J.; Wang, Z.; Sun, W.; Pan, J.; Liu, J.; Pang, J.; Lin, L. LSTM Pose Machines. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.

9. Dutson, M.; Li, Y.; Gupta, M. Event Neural Networks. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
10. Habibian, A.; Abati, D.; Cohen, T.S.; Bejnordi, B.E. Skip-Convolutions for Efficient Video Processing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
11. Parger, M.; Tang, C.; Twigg, C.D.; Keskin, C.; Wang, R.; Steinberger, M. DeltaCNN: End-to-End CNN Inference of Sparse Frame Differences in Videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
12. Parger, M.; Tang, C.; Neff, T.; Twigg, C.D.; Keskin, C.; Wang, R.; Steinberger, M. MotionDeltaCNN: Sparse CNN Inference of Frame Differences in Moving Camera Videos with Spherical Buffers and Padded Convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, 1–6 October 2023.
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
15. Xu, Y.; Zhang, J.; Zhang, Q.; Tao, D. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
16. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision Transformers for Dense Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
17. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
18. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
19. Zhu, X.; Dai, J.; Yuan, L.; Wei, Y. Towards High Performance Video Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
20. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
21. Xu, M.; Xiong, Y.; Chen, H.; Li, X.; Xia, W.; Tu, Z.; Soatto, S. Long Short-Term Transformer for Online Action Detection. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 6–14 December 2021.
22. Yang, L.; Han, J.; Zhang, D. Colar: Effective and Efficient Online Action Detection by Consulting Exemplars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
23. Wang, X.; Zhang, S.; Qing, Z.; Shao, Y.; Zuo, Z.; Gao, C.; Sang, N. OadTR: Online Action Detection with Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
24. Zhao, Y.; Krähenbühl, P. Real-Time Online Video Detection with Temporal Smoothing Transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
25. Song, J.; Wang, L.; Gool, L.V.; Hilliges, O. Thin-Slicing Network: A Deep Structured Model for Pose Estimation in Videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
26. Nie, X.; Li, Y.; Luo, L.; Zhang, N.; Feng, J. Dynamic Kernel Distillation for Efficient Pose Estimation in Videos. In Proceedings of the ICCV 2019, Seoul, Republic of Korea, 27 October–2 November 2019.
27. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by Tracking: Siamese CNN for Robust Target Association. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016.
28. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.T.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the ICIP 2016, Phoenix, AZ, USA, 25–28 September 2016.
29. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the ICIP 2017, Beijing, China, 17–20 September 2017.
30. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-object Tracking by Associating Every Detection Box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
31. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
32. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
33. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
34. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
35. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. Vivit: A video vision transformer. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.

36. Bertasius, G.; Wang, H.; Torresani, L. Is space-time attention all you need for video understanding? In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021.
37. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021.
38. Piergiovanni, A.; Angelova, A.; Ryoo, M.S. Tiny video networks. *Appl. AI Lett.* **2022**, *3*, e38. [CrossRef]
39. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
40. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
41. Luo, J.; Wu, J.; Lin, W. ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
42. Wu, S.; Li, G.; Chen, F.; Shi, L. Training and Inference with Integers in Deep Neural Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
43. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.G.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
44. Teerapittayanon, S.; McDanel, B.; Kung, H.T. BranchyNet: Fast inference via early exiting from deep neural networks. In Proceedings of the International Conference on Pattern Recognition, Cancún, Mexico, 4–8 December 2016.
45. Figurnov, M.; Collins, M.D.; Zhu, Y.; Zhang, L.; Huang, J.; Vetrov, D.P.; Salakhutdinov, R. Spatially Adaptive Computation Time for Residual Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
46. Wu, Z.; Nagarajan, T.; Kumar, A.; Rennie, S.; Davis, L.S.; Grauman, K.; Feris, R. Blockdrop: Dynamic inference paths in residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
47. Lin, J.; Rao, Y.; Lu, J.; Zhou, J. Runtime Neural Pruning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
48. Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; Xie, P. EViT: Expediting Vision Transformers via Token Reorganizations. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022.
49. Meng, L.; Li, H.; Chen, B.C.; Lan, S.; Wu, Z.; Jiang, Y.G.; Lim, S.N. Adavit: Adaptive vision transformers for efficient image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
50. Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; Hsieh, C.J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 6–14 December 2021.
51. Yin, H.; Vahdat, A.; Alvarez, J.M.; Mallya, A.; Kautz, J.; Molchanov, P. A-vit: Adaptive tokens for efficient vision transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
52. Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; Chen, L. FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
53. Kim, H.; Jain, M.; Lee, J.T.; Yun, S.; Porikli, F. Efficient action recognition via dynamic knowledge propagation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
54. Meng, Y.; Lin, C.C.; Panda, R.; Sattigeri, P.; Karlinsky, L.; Oliva, A.; Saenko, K.; Feris, R. Ar-net: Adaptive frame resolution for efficient action recognition. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
55. Sun, X.; Panda, R.; Chen, C.F.R.; Oliva, A.; Feris, R.; Saenko, K. Dynamic network quantization for efficient video inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
56. Cavigelli, L.; Benini, L. CBinfer: Exploiting Frame-to-Frame Locality for Faster Convolutional Network Inference on Video Streams. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 1451–1465. [CrossRef]
57. De Alwis, U.; Alioto, M. TempDiff: Temporal Difference-Based Feature Map-Level Sparsity Induction in CNNs with <4% Memory Overhead. In Proceedings of the 3rd IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2021, Washington, DC, USA, 6–9 June 2021.
58. Pan, B.; Lin, W.; Fang, X.; Huang, C.; Zhou, B.; Lu, C. Recurrent residual module for fast inference in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
59. Choi, J.; Lee, S.; Chu, J.; Choi, M.; Kim, H.J. vid-TLDR: Training Free Token merging for Light-weight Video Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024.
60. Park, S.H.; Tack, J.; Heo, B.; Ha, J.; Shin, J. K-centered Patch Sampling for Efficient Video Recognition. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022. [CrossRef]
61. Chen, L.; Tong, Z.; Song, Y.; Wu, G.; Wang, L. Efficient Video Action Detection with Token Dropout and Context Refinement. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 1–6 October 2023.

62. Wu, Q.; Cui, R.; Li, Y.; Zhu, H. HaltingVT: Adaptive Token Halting Transformer for Efficient Video Recognition. *arXiv* **2024**, arXiv:2401.04975.
63. Peng, H.; Feng, W.; Li, H.; Zhan, Y.; Zhou, Q.; Xia, Y. Arena: A Patch-of-Interest ViT Inference Acceleration System for Edge-Assisted Video Analytics. *arXiv* **2024**, arXiv:2404.09245.
64. Liu, Y.; Matsoukas, C.; Strand, F.; Azizpour, H.; Smith, K. Patchdropout: Economizing vision transformers using patch dropout. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023.
65. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
66. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
67. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
68. Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video Swin Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
69. Ozaki, Y.; Tanigaki, Y.; Watanabe, S.; Onishi, M. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. In Proceedings of the GECCO'20: Genetic and Evolutionary Computation Conference, Cancún, Mexico, 8–12 July 2020.
70. Stein, S.; McKenna, S.J. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In Proceedings of the The ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013.
71. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
72. Li, F.-F.; Deng, J.; Li, K. ImageNet: Constructing a large-scale image database. *J. Vis.* **2009**, *9*, 1037.
73. Farha, Y.A.; Gall, J. MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
74. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
75. Yang, Y.; Ramanan, D. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2878–2890. [CrossRef] [PubMed]
76. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V*; Springer: Cham, Switzerland, 2014.
77. Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv* **2017**, arXiv:1704.00675.
78. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
79. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In Proceedings of the International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016.
80. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020.
81. Yao, Z.; Cao, Y.; Lin, Y.; Liu, Z.; Zhang, Z.; Hu, H. Leveraging Batch Normalization for Vision Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Montreal, QC, Canada, 10–17 October 2021.
82. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
83. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
84. Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; Black, M.J. Towards understanding action recognition. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013.
85. Dai, Z.; Heckel, R. Channel Normalization in Convolutional Neural Network avoids Vanishing Gradients. *arXiv* **2019**, arXiv: 1907.09539.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

High-Precision Multi-Object Tracking in Satellite Videos via Pixel-Wise Adaptive Feature Enhancement

Gang Wan [1,2], Zhijuan Su [1,*], Yitian Wu [1], Ningbo Guo [1], Dianwei Cong [1,2], Zhanji Wei [1], Wei Liu [1] and Guoping Wang [1]

[1] School of Space Information, Space Engineering University, Beijing 101407, China; dsddfff@stu.cuz.edu.cn (G.W.); ytwu@whu.edu.cn (Y.W.); sxguonb@163.com (N.G.); congdianwei@sina.com (D.C.); weizhanji@163.com (Z.W.); ggfh@stu.cuz.edu.cn (W.L.); m18519338782@163.com (G.W.);

[2] State Key Laboratory of Geo-Information Engineering, Xi'an 710054, China

* Correspondences: hgfgh@stu.cuz.edu.cn

Abstract: In this paper, we focus on the multi-target tracking (MOT) task in satellite videos. To achieve efficient and accurate tracking, we propose a transformer-distillation-based end-to-end joint detection and tracking (JDT) method. Specifically, (1) considering that targets in satellite videos usually have small scales and are shot from a bird's-eye view, we propose a pixel-wise transformer-based feature distillation module through which useful object representations are learned via pixel-wise distillation using a strong teacher detection network; (2) targets in satellite videos, such as airplanes, ships, and vehicles, usually have similar appearances, so we propose a temperature-controllable key feature learning objective function, and by highlighting the learning of similar features during distilling, the tracking accuracy for such objects can be further improved; (3) we propose a method that is based on an end-to-end network but simultaneously learns from a highly precise teacher network and tracking head during training so that the tracking accuracy of the end-to-end network can be improved via distillation without compromising efficiency. The experimental results on three recently released publicly available datasets demonstrated the superior performance of the proposed method for satellite videos. The proposed method achieved over 90% overall tracking performance on the AIR-MOT dataset.

Keywords: satellite video; transformer attention learning; multi-object tracking (MOT)

1. Introduction

It is of great importance to process remote sensing images, especially for Earth observation systems [1–6]. In recent years, with the development of remote sensing technology, satellite videos have emerged as a new type of remote sensing data. The interpretation of satellite videos has important applications in commercial monitoring, military surveillance, natural disaster prediction, and so on [7–10]. This paper aims to tackle the multi-object tracking (MOT) task for satellite videos. The MOT task refers to detecting multiple objects in a video and assigning unique IDs for objects across frames without knowing the number of objects in advance. Recently, researchers have proposed many MOT methods. The most widely used of these methods can be divided into two categories: detection-by-tracking (DBT) [11–16] and joint detection and tracking (JDT) [17–22].

DBT methods first adopt a object detector to generate object detection bounding boxes in each frame via object detection methods, such as Faster R-CNN [23], SDP [24], DPM [25], etc., and then use a data association approach to associate the detection results and obtain the trajectories. SORT [11] is the most classic DBT method, which is mainly composed of object detection, a Kalman filter [26], and the Hungarian algorithm [27]. DeepSORT [13] improves SORT by introducing a deep learning model to extract appearance features for closest proximity matching during real-time tracking. Then, StrongSORT [15] further

improves the method in various aspects of object detection, feature representation, and matching. In addition, ByteTrack [12], GHOST [28], IOU-Tracker [16], and so on have also been proposed and follow the DBT framework. DBT-based methods usually have high tracking accuracy, which benefits from specialized and separate detection and data association methods. However, using these two separate steps results in relatively low tracking efficiency.

JDT methods achieve the MOT task via multi-task learning in an end-to-end network, namely by simultaneously accomplishing object detection and data association tasks. CenterTrack [19], which is based on CenterNet [29], inputs two consecutive frames and a heat map of the previous frame and directly outputs the location, size, confidence score, and ID of targets. The goal is to detect and track objects simultaneously via a shared feature learning network. Based on CenterNet, FairMOT [21] estimates target centers and positions on high-resolution feature maps and introduces parallel branches to estimate pixel-level Re-ID features. TraDeS [22] estimates the offsets of tracked targets based on cost amounts. TrackFormer [30] conducts joint object detection and tracking via transformer-based attention that connects past and future frames. With shared networks, JDT methods can usually track objects efficiently. However, due to their use of point-based detection, their tracking results are often less stable and accurate than those of DBT methods. A comparison between DBT and JDT methods is summarized in Table 1.

Table 1. Comparison between DBT and JDT methods.

	Flowchart	**Properties**
DBT	Object detection + data association	High detection accuracy; Low tracking efficiency
JDT	End-to-end	Feature sharing; High tracking efficiency; Low stability
Proposed	End-to-end + distillation	Feature sharing; High tracking efficiency; High detection accuracy

The DBT and JDT methods have complementary properties. If we use an end-to-end network for tracking while training it with both an object detection network and a tracking head, then object representation can be enhanced without influencing tracking efficiency. As a consequence, in this paper, we propose a transformer distillation-based end-to-end joint detection and tracking framework, which is based on pixel-wise feature distillation and employs a highly precise object detection network as a teacher. In this way, based on an end-to-end network, distilled features are expected to improve the performance of object localization by enhancing key object features. Meanwhile, since the distillation is only implemented in the training process, the tracking efficiency of the tracking network will not be influenced. Considering that targets in satellite videos usually have small scales and are shot from a bird's-eye view, we propose a pixel-wise transformer-based feature distillation module in order to distill more detailed features from the strong teacher network. Moreover, to distill critical features, we propose a temperature-controllable key feature learning objective function.

As a consequence, based on the principle of improving detection capability of the end-to-end tracking network via distillation without influencing the tracking efficiency, we propose a new transformer distillation-based JDT method and design a corresponding distillation method. During training, it minimizes not only the tracking losses, but also the proposed feature distillation loss. The experimental results demonstrated the effectiveness and efficiency of the proposed method, which could significantly surpass the current state-of-the-art methods.

In general, the contributions of this paper are mainly reflected in the following four aspects:

(1) To meet both the efficiency and accuracy requirements of applications in satellite videos, we propose a transformer distillation-based end-to-end joint detection and tracking (JDT) method for the MOT task. It inherits the efficiency of a JDT-based framework in an end-to-end network during tracking. Meanwhile, it also learns effective object representation from a detection network via a DBT-based framework.

(2) Considering that targets in satellite videos usually have small scales and are shot from a bird's-eye view, we propose a pixel-wise transformer distillation module to improve tracking accuracy by distilling pixel-wise detail features from a strong teacher network.

(3) Considering that targets in satellite videos usually have similar appearances, we propose a temperature-controllable feature distillation objective function to further enhance the distillation of the useful and critical features of similar targets.

(4) Experimental results on three publicly available datasets (AIR-MOT, SAT-MTB, and VISO) showed that the performance of the proposed method is superior to the baseline method and significantly outperforms the current state-of-the-art methods.

2. Related Work

2.1. MOT in Satellite Videos

Due to the significance of MOT in a range of applications, many meaningful MOT methods for satellite videos have recently been proposed. Unlike those in nature videos, the targets in satellite videos are usually small and have indistinguishable features. Therefore, motion information is especially important for MOT. In [31], a dataset named VISO was constructed, which mainly focuses on the detection of moving objects in MOT. Moreover, a motion modeling method and a novel detection method were also proposed in [31]. The motion modeling method aims to integrate motion information in MOT by accumulative multi-frame differencing and matrix completion, while the detection method works by modeling local noise via exponential probability distribution with a multi-morphological cue. Focusing on tracking vehicles that are crowded and have similar appearances, a DBT method was proposed in [32] that integrates complementary information between adjacent frames via a customized cross-frame object detection network and captures motion information via a two-branch architecture based on long short-term memory. Detection performance significantly influences the tracking accuracy of MOT. In [33], based on a single-object tracking method, a bidirectional framework was proposed that reduces the degradation of detection networks. Moreover, SFMFMOT, which was proposed in [34], fully considers low-confidence bounding boxes, missed objects, false alarms, and ID switches by exploring slow and motion features. For satellite-based applications, efficiency is also important. A JDT method named TGraM was proposed in [10], which approaches MOT as a graph reasoning procedure and uses a multi-task learning framework. In this paper, considering both the efficiency and accuracy requirements of MOT in satellite videos, we propose a transformer distillation-based end-to-end joint detection and tracking method that learns accurate detection from a teacher network that is based on a JDT baseline.

2.2. Transformer-Based MOT Methods

A transformer [35] is a type of neural network that learns contexts and meanings by tracking relationships in sequence data. In recent years, researchers have proposed many multi-object tracking methods based on transformers. GTR [36] takes a short sequence of frames as input and produces global trajectories for all objects. MeMOT [37] employs an online tracking algorithm that performs object detection and data association under a common framework, which is capable of linking objects after a long timespan. MOTR [38] extends DETR [39] and introduces track queries to model the tracked instances in an entire video. It also uses a tracklet-aware label assignment method to train and track newborn object queries, a temporal aggregation network, and a collective average loss to enhance temporal relation modeling. MOTRv2 [40] improves MOTR by elegantly incorporating an extra object detector. P3AFormer [41] tracks objects as pixel-wise distributions. TrackFormer [30] formulates the MOT task as a frame-to-frame set prediction problem based

on a transformer. In this paper, to effectively distill object features in satellite videos, we propose a pixel-wise transformer distillation module for accurate MOT in satellite videos.

2.3. Knowledge Distillation for MOT

Knowledge distillation [42] is a popular technique that is widely used in many learning tasks, such as facial recognition [43], semantic segmentation [44], behavior prediction [45], and tracking, including single-object tracking [46] and MOT [47]. Knowledge distillation trains a student network not only from data but also from highly precise teacher networks, thereby enhancing the performance of the student network. For MOT, the method proposed in [47] focuses on association performance and distills Re-ID information.

In this paper, we focus on the detection performance of a JDT-based method. The proposed method is different from existing knowledge-distillation-based MOT methods in two ways: first, we propose a transformer-based pixel-wise distillation module, which benefits from learning the features of small targets in satellite videos; second, based on the features of transformer distillation, we propose a temperature-controllable key feature learning objective function, which further improves the feature learning performance of similar targets and tracking performance.

3. The Proposed Method

In this study, in order to illustrate the proposed method, we chose a classic, state-of-the-art end-to-end tracking network named CenterTrack [19] as the baseline method. The teacher model we chose was a pre-trained, highly precise object detection network named YOLOX. On this basis, we designed a pixel-wise transformer distillation module to distill object features from YOLOX. Then, to better learn the key features of similar objects, we designed a temperature-controllable key feature learning objective function. In the following sections, we first briefly review CenterTrack, the pixel-wise transformer distillation module, the temperature-controllable key feature learning objective function, and the corresponding optimization derivatives and tracklet generation.

3.1. CenterTrack

CenterTrack [19] is an end-to-end JDT framework. The inputs for the network are the current frame $I^{(t)} \in \mathbb{R}^{H\times W\times 3}$, the last frame $I^{(t-1)} \in \mathbb{R}^{H\times W\times 3}$, and the tracked objects in the last frame $T^{(t-1)} = \{b_0^{(t-1)}, b_1^{(t-1)}, \ldots\}$, which are represented in a heat map. During tracking, at frame t, the objects detected by CenterTrack are denoted by $T^{(t)} = \{b_0^{(t)}, b_1^{(t)}, \ldots\}$, and their attributes are denoted by $b = (\boldsymbol{p}, \boldsymbol{s}, w, id)$, where $\boldsymbol{p} \in \mathbb{R}^2$ is the center location, $\boldsymbol{s} \in \mathbb{R}^2$ represents size, $w \in [0, 1]$ reflects detection confidence, and $id \in \mathbb{I}$ is the unique identity. To train the network, the overall objective L^T should be minimized, including heat map loss, which highlights the center location of targets using the following formula:

$$L_H = \frac{1}{N}\begin{cases}(1-\hat{Y})^2\log(\hat{Y}) & Y = 1\\ (1-Y)^4\hat{Y}^2\log(1-\hat{Y}) & otherwise\end{cases} \tag{1}$$

where Y and $\hat{Y}$ denote the predicted and ground-truth heat maps with N objects, respectively. Size prediction loss is calculated as follows:

$$L_S = |Size(\boldsymbol{p}) - \boldsymbol{s}| \tag{2}$$

The offset and center location loss are calculated as follows:

$$L_O = |(\hat{\boldsymbol{p}}^{t-1} - \hat{\boldsymbol{p}}^t) - (\boldsymbol{p}^{t-1} - \boldsymbol{p}^t)| \tag{3}$$

However, in satellite videos, the targets are usually small and have highly similar appearances, which results in confused object localization. As a consequence, in this paper,

we propose a pixel-wise transformer distillation module and a temperature-controlled key feature learning objective function to distill object information and increase tracking accuracy in a JDT-based framework while maintaining high tracking efficiency. In this study, CenterTrack was used as the baseline, and we used knowledge distillation to distill detection knowledge from YOLOX into CenterTrack.

3.2. Overall Architecture

The tracking pipeline of the proposed method is shown in Figure 1. The overall framework is composed of a teacher network (a pre-trained object detection network used to distill critical features during training) and a JDT-based student network, which is used for generating ground boxes and trajectories during tracking. The flowchart of training and test are also illustrated. For the $t-$frame, the inputs of CenterTrack are $I^{(t)}$, $I^{(t-1)}$, and $T^{(t-1)}$, and the feature map F^s is obtained through the backbone network ϕ_T as follows:

$$F^s = \phi_T(I^{(t)}) \tag{4}$$

At the same time, the feature maps $\{F_1^t, F_2^t, F_3^3\}$ of image $I^{(t)}$ are obtained through the object detection backbone network ϕ_D as follows:

$$\{F_1^t, F_2^t, F_3^3\} = \phi_D(I^{(t)}) \tag{5}$$

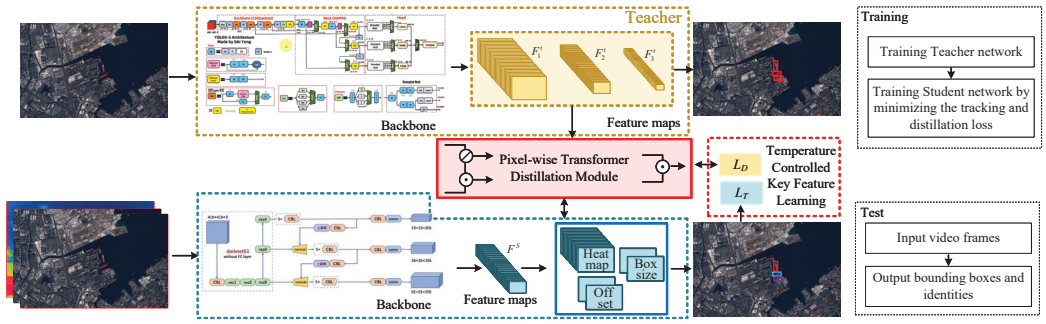


Figure 1. The overall pipeline and flowchart of the proposed method.

We regard the features output by the object detection network as *Key* (K) features and the features obtained by the tracking network as *Query* (Q) features. Since targets in satellite videos are usually small and taken from a bird's-eye view, we use transformer-based pixel-wise attention to distill *Key* features. Considering that targets in satellite videos also usually have similar appearances, we added a temperature-controllable key-feature-learning-based objective function L_D to L_T, as in CenterTrack, for optimization to obtain the final results. The pixel-wise transformer distillation and temperature-controlled key feature learning objective are introduced in the following sections.

3.3. Pixel-Wise Transformer Distillation

To distill the object information in the baseline, i.e., CenterTrack, we designed a pixel-wise transformer distillation architecture, as shown in Figure 2. Before distillation learning, a highly precise teacher detection network ϕ_D is first pre-trained, as shown in Figure 1. Then, the multi-scale features F^t extracted from the network can effectively represent the semantic features of the corresponding targets. As a consequence, if these features can be well distilled in the JDT-based tracking network, its detection performance is expected to improve without impacting tracking efficiency.

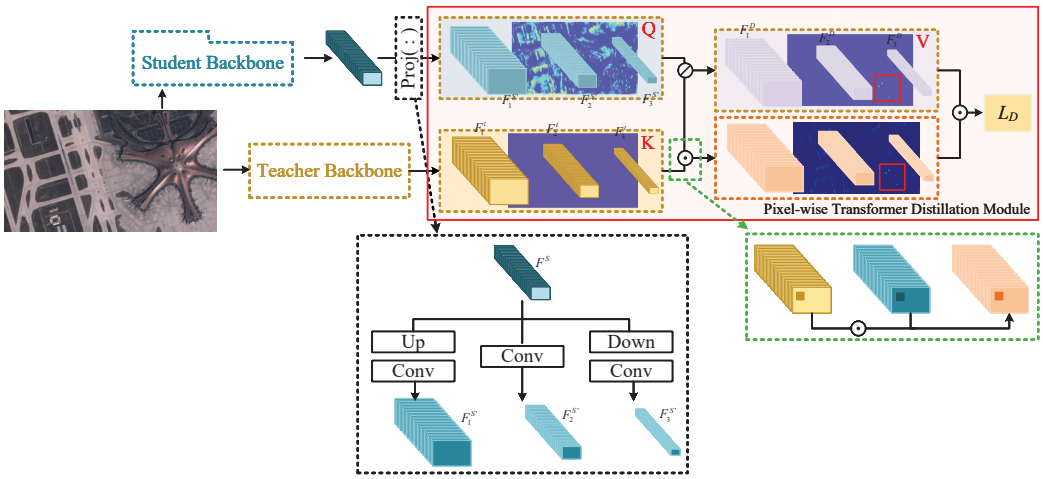


Figure 2. Pixel-wise transformer distillation.

During training, with the training frame $I^{(t)}$, the highly precise object features F_1^t, F_2^t, and F_3^t at different scales are first extracted via ϕ_D. Meanwhile, $I^{(t)}$ is fed into the CenterTrack network ϕ_T, and the tracking feature F^s is then generated and fed into two branches: one is the tracking head to generate tracking loss for optimization; the other is for object feature distillation. To distill multi-scale features, considering the scale differences between F^s and F_1^t, F_2^t, and F_3^t, F^s is first projected into multi-scale feature maps $F_1^{s'}$, $F_2^{s'}$, and $F_3^{s'}$ via several convolutional layers, as follows:

$$F_1^{s'} = Conv_{1\times1}(Conv_{3\times3}(Conv_{1\times1}(F^s))) \tag{6}$$

$$F_2^{s'} = Conv_{1\times1}(Conv_{3\times3}(Conv_{1\times1}(F_1^{s'}))) \tag{7}$$

$$F_3^{s'} = Conv_{1\times1}(Conv_{3\times3}(Conv_{1\times1}(F_2^{s'}))) \tag{8}$$

Then, the feature maps of F_1^D, F_2^D, and F_3^D are constructed as follows:

$$F_1^D = F_1^t - F_1^{s'} \tag{9}$$

$$F_2^D = F_2^t - F_2^{s'} \tag{10}$$

$$F_3^D = F_3^t - F_3^{s'} \tag{11}$$

The aim of distillation is to minimize differences; however, tracking performance may decrease due to irrelevant features from the detection network. As a consequence, we constructed a pixel-wise transformer distillation module.

To distill tracking-related features, the features of K and Q are first constructed as F^t and $F^{s'}$, respectively. Then, the final pixel-wise transformer distillation-based difference $F_i^{D'}$ is generated via transformer-based attention as follows:

$$F_1^{D'} = Softmax(F_1^{s'} \odot F_1^t) \odot F_1^D \tag{12}$$

$$F_2^{D'} = Softmax(F_2^{s'} \odot F_2^t) \odot F_2^D \tag{13}$$

$$F_3^{D'} = Softmax(F_3^{s'} \odot F_3^t) \odot F_3^D \tag{14}$$

where $\odot$ denotes the element-wise multiplication operator.

Using the transformer-based module, object features relevant to tracking can be specifically highlighted and distilled. Moreover, the pixel-wise transformer distillation module is implemented at different scales, so the features of both large and small objects

can be extracted, and feature distillation effectiveness can be improved. For the pixel-wise transformer distillation module, the distillation objective function is defined based on $F^{D'}$.

3.4. Temperature-Controlled Key Feature Learning

In contrast to existing feature distillation methods, which usually directly minimize the distance between two features, we aimed to distill critical target features and restrain background features in distillation. As a consequence, we designed a new distillation objective and introduced a distance temperature T to control the generalization of distilled features. This objective is used to highlight target information and distill critical features.

Using the tracking loss L^T defined in CenterTrack, we developed a temperature-controllable feature distillation loss L^D by introducing temperature T as follows:

$$L_1^D = exp(F_1^{D'} \odot F_1^D / T) \tag{15}$$

$$L_2^D = exp(F_2^{D'} \odot F_1^D / T) \tag{16}$$

$$L_3^D = exp(F_3^{D'} \odot F_1^D / T) \tag{17}$$

The total distillation loss L^D is the sum of L_1^D, L_2^D, and L_3^D:

$$L^D = L_1^D + L_2^D + L_3^D \tag{18}$$

Here, T controls the variance among different feature values in the feature maps. Lower T values result in a larger variance between small and large feature values, which can highlight more critical objects and significantly restrain background objects. Meanwhile, larger T values result in lower variance, and background information can also be distilled into the tracking network. With larger T values, the feature maps highlight many background objects, while lower T values filter out most objects. Larger T values focus on more objects, and more targets can be distilled. With this controllable temperature, useful features for tracking can be effectively distilled.

Then, with the tracking loss, the final objective is defined as follows:

$$L = L^T + \lambda L^D \tag{19}$$

where λ is the user-defined parameter that controls the weights of the two terms.

3.5. Optimization

With the above objective, the back-propagation process can be derived. For the distillation objective, the gradient of the output is first computed as follows:

$$\delta F_i^{D'} = \frac{\partial L_D}{\partial F_i^D} = \exp(F^{D'} \odot F^D / T) \odot F^{D'} \tag{20}$$

From $\delta F_i^{D'}$, the gradient of $F_i^{s'}$ can be easily derived as follows:

$$\begin{aligned} \delta F_i^{s'} =& F_i^D F_i^t \Delta Softmax(F_i^{s'} \odot F_i^t) \\ & - Softmax(F_i^{s'} \odot F_i^t) \end{aligned} \tag{21}$$

where $\Delta Softmax(\cdot)$ is the gradient of the Softmax activation function. Then, the gradient $\delta F_i^{s'}$ back-propagated through the projection operators and δF^s is obtained, along with the back-propagated gradient from the tracking head.

After training, the test process follows CenterTrack, which maintains high tracking efficiency. Meanwhile, with the object feature distillation, the network inherits the detection performance of an accurate detection network. Moreover, the proposed distillation objective function focuses more on targets and, thus, is able to reduce the negative effects of feature distillation.

3.6. Tracklet Generation

For each input video frame, we output the center positions of all predicted targets and the prediction biases of the center points versus those in the previous frame. We then match the detected positions and the trajectories that are closest to those positions. At the same time, considering the small target position changes between adjacent frames, we limit the matching distance via the geometric mean of the length and width of the prediction boxes to avoid false matching over long periods. However, there is still a problem in that the prediction accuracy of targets with different sizes, moving speeds, and directions cannot be guaranteed with only one bias prediction head. For example, large prediction biases can be obtained for slow-moving objects with high frame rates or stationary targets, which are far from the real trajectories. As a consequence, we further adopt a simple and effective intersection over union (IOU)-based secondary matching method. Unmatched targets and trajectories from the first matching process are matched again, only according to the IOU scores between them. Meanwhile, to avoid false matches over long periods, we limit the successfully matched targets and trajectories that achieve IOU scores over 0.5. For trajectory initialization and updates, we use the settings in CenterTrack.

In addition, even with pixel-wise transformer distillation and temperature-controllable distillation, the distilled knowledge cannot be guaranteed to be completely correct. For example, due to the complex backgrounds of satellite videos, many background objects with similar appearances to targets could be wrongly recognized, which increases the number of false detections. Fortunately, due to the low confidence of these false detections, we found that they were usually temporally detected in one or two frames. As a consequence, we designed a 3-frame-based decision strategy. Different from DeepSORT, which completely abandons mismatched detections in two contiguous frames regardless of possible successful matches in subsequent frames, we defined a list of three components that records the matching status in three consecutive frames for each trajectory. If the detected targets successfully match the trajectories over three consecutive frames, then the three matches are determined and output. Otherwise, the targets are taken as false detections and abandoned. This strategy effectively reduces the influence of false detections on precision. Moreover, we remove trajectories that are unmatched over 30 consecutive frames and do not consider them as potential trajectories in subsequent frames.

4. Experimental Study

4.1. Dataset Description

Experiments were conducted by comparing our proposed method to a baseline and other current state-of-the-art MOT methods on three new and publicly available datasets, including AIR-MOT [10], SAT-MTB [48], and VISO [31].

The AIR-MOT [10] dataset is composed of 59 satellite videos, acquired by the Jilin-1 satellite. They dynamically recorded scenes from the airports and harbors in the cities of Abu Dhabi, Dubai, San Diego, Los Angeles, Sydney, Yokohama, Beijing, Sanya, and Shanghai from October 2017 to October 2020. Therefore, the annotated targets include aircraft and ships. In our experiments, we evaluated the methods separately on different targets, which were divided into two subsets, named AIR-aircraft and AIR-ship. Regarding the videos in the dataset, the size of the frames is 1920×1080 pixels and the spatial resolution ranges from 0.91 m to 1.27 m. The frame rate of the videos is 5–10 frames-per-second (FPS). In this dataset, aircraft and ships have different properties. Aircraft are usually large, sparsely arranged, and have distinguishable features, while ships are much smaller and densely arranged.

The videos in the SAT-MTB [48] dataset were also acquired by Jilin-1 but covered much larger regions across the USA, Mexico, Australia, Spain, Italy, France, Turkey, Japan, India, Thailand, Saudi Arabia, the United Arab Emirates, and China, with a spatial resolution of 0.92 m. The acquisition time was from 2017 to 2021, and there are 249 videos containing annotated targets in this dataset. The frame sizes of these videos range from 512×512 to

1080 × 2160 pixels, and the frame rate is about 10 FPS. The targets in this dataset were also divided into two subsets, named SAT-MTB-airplane and SAT-MTB-ship.

The VISO [31] dataset contains 47 videos that include annotated targets. They were also acquired by the Jilin-1 satellite, with a spatial resolution of 0.91 m. The frame sizes of these videos range from 1024 × 1024 to 1025 × 1025 and the frame rate is about 10 FPS. The targets in this dataset are mainly cars, which are also small and densely arranged. Some targets and scenes from the three datasets are illustrated in Figure 3.

For the division of training and test sets, we followed the standard training and test ratios of 1:1, 3:2, and 3:7 for the AIR-MOT, SAT-MTB, and VISO datasets, respectively, for fair comparison. As shown in the figures, the targets in these satellite videos were usually small, and the smallest size of detectable objects was 20 × 21 pixels.

Figure 3. Some targets and images from the three datasets.

4.2. Evaluation Metrics

We used fragmentation (FM), false positives (FPs), false negatives (FNs), mostly lost (ML), mostly tracked (MT), identification recall (IDR), identification precision (IDP), ID

switches (IDs), identification F-score (IDF1), and multi-object tracking accuracy (MOTA) as the evaluation criteria, as do many satellite video MOT methods [10]. Among them, MOTA is a comprehensive evaluation metric that measures the overall accuracy of tracking as follows:

$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FP} + \mathrm{FN} + \mathrm{IDs})}{\sum_t \mathrm{GT}} \tag{22}$$

where t denotes the frame index and GT denotes the number of ground-truth bounding boxes in each frame. Larger MOTA values mean higher tracking accuracy, with 1 as the best case. IDF1 denotes the associated ratio among detected and tracked objects and is calculated as follows:

$$\mathrm{IDF1} = \frac{2\mathrm{IDTP}}{2\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{23}$$

where IDTP and IDFP represent the number of true IDs and false positive IDs, respectively, and IDFN is the number of false negative IDs. IDF1 stands for identification F-Scorel, and larger IDF1 values indicate more accurate associations. IDP and IDR denote the precision and recall, respectively, of ID tracking. MT and ML represent the ratio of correctly tracked trajectories in over 80% and below 20% of all frames, respectively. FM denotes the cut-off time of a trajectory. IDs represents the number of identification switches.

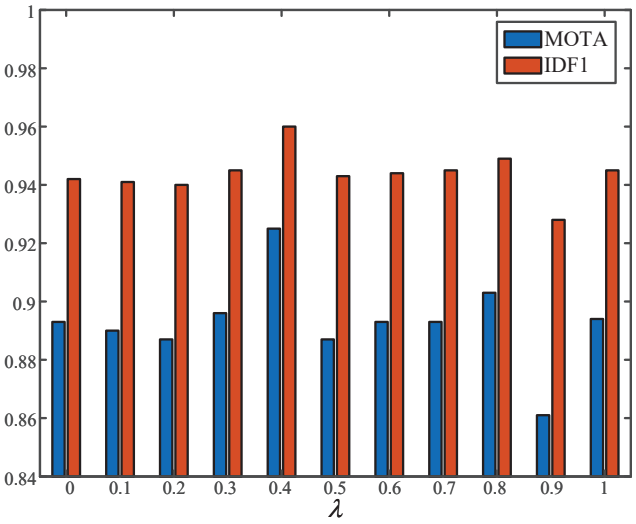
4.3. Implementation Details

Without loss of generality, in our experiments, we chose YOLOX [49] as the highly precise object detection network that was used to guide the tracking network, i.e., the baseline (CenterTrack), for feature distillation learning. In order to avoid errors caused by feature map scale mismatches during feature learning, for both networks, we resized the input images during training to 640×640. In addition, since our knowledge distillation module could be ignored during testing, we resized the images to 1088×608 for higher accuracy, which was more similar to the original scale of the images. Following CenterTrack's original settings, we trained the model using the Adam optimizer for 70 epochs, accompanied by an initial learning rate of 1.25×10^{-4} and scaled down at 60 epochs by 10. We set the temperature to $T = 50$ and the distillation weights λ to 0.2 and 0.4 for AIR-ship and AIR-aircraft targets, respectively. Considering that the partial AIR-ship dataset had an average of more than 100 targets per frame, we modified the maximum number of targets from 100 to 150 for this dataset. The other model parameters followed CenterTrack's default settings, and more experimental details can be found in CenterTrack.

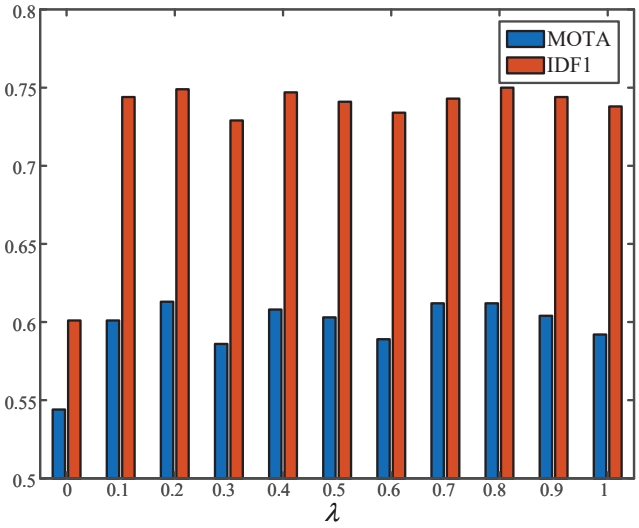
4.4. Evaluation of λ

Since the characteristics of airplanes and ships are different in the AIR-MOT dataset, we tested the most suitable λ on two partial datasets separately. We selected the two most important indicators out of the multi-objective tracking evaluation metrics, namely, MOTA and IDF1, as shown in Figure 4. Figure 4a shows the results for the AIR-aircraft dataset and Figure 4b shows the results for the AIR-ship dataset.

As shown in Figure 4, the most suitable λ was 0.2. When the value of λ was 0.2, the MOTA value reached 0.6140 and the IDF1 value reached 0.7430. Therefore, in the subsequent experiments, the λ for the AIR-aircraft dataset was set to 0.2. As shown in Figure 4, when λ was 0.4, the MOTA value reached 0.9270 and the IDF1 value reached 0.9610. Therefore, for the subsequent experiments, the λ for the AIR-ship dataset was set to 0.4.



(**a**) AIR-aircraft targets



(**b**) AIR-ship targets

Figure 4. Evaluation of λ on AIR-ship and AIR-aircraft targets.

4.5. Ablation Studies

We selected CenterTrack as the baseline method to conduct ablation experiments on the AIR-aircraft and AIR-ship datasets. The visualized results are illustrated in Figure 5. The quantified comparison results are shown in Tables 2 and 3. The "Tracklets" label represents the tracklet generation approach designed in this study, and the "Transformer" label refers to the transformer distillation module proposed in this paper. As can be seen from Table 2, the proposed method exceeded the baseline MOTA by about 7%. In terms of IDF1, the proposed method exceeded the baseline by 14.2%. In addition, in terms of FM, ML, MT, IDR, IDP, and IDs, the proposed method improved on the baseline by 85, 8, 14, 13.8%, 16.9%, and 2411, respectively. This demonstrates that the proposed approach is obviously superior to the baseline method.

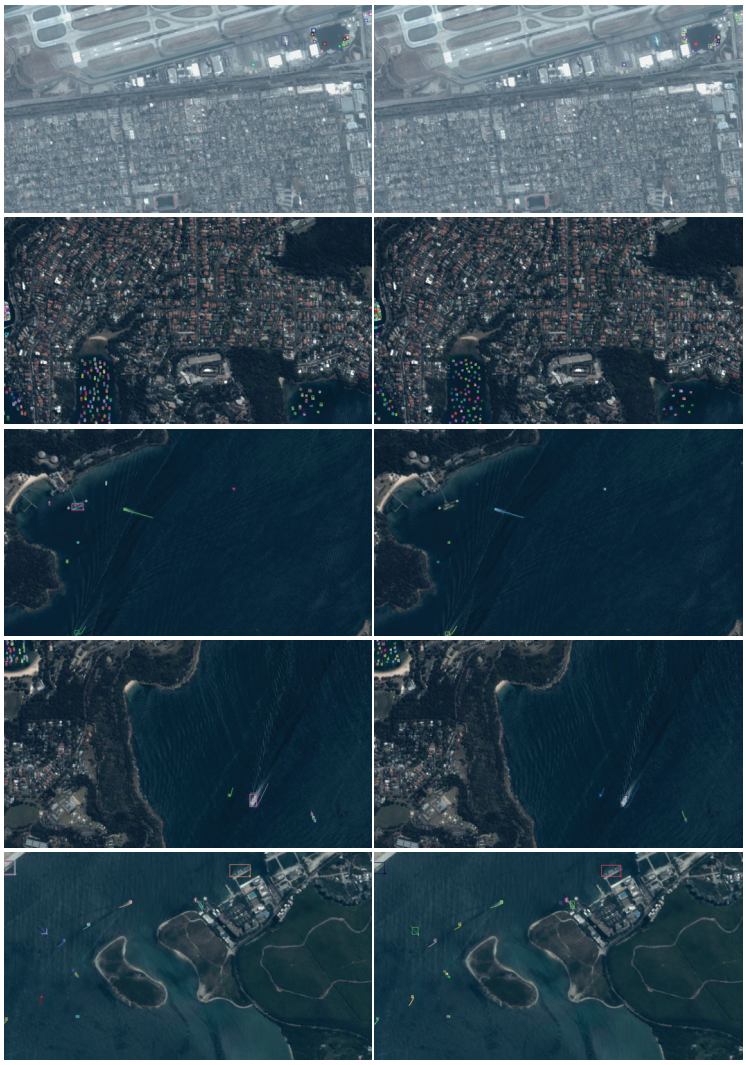


Figure 5. Visualization of several tracking results using the baseline method (CenterTrack) and the proposed method.

Table 2. Results of ablation experiments on AIR-ship targets.

Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
Baseline	640	143	297	52.40%	70.30%	2461	60.10%	54.40%
Baseline + Tracklets	638	141	292	62.65%	84.40%	71	72.00%	56.90%
Baseline + Transformer	701	143	291	56.90%	79.50%	234	67.30%	59.39%
Proposed Method	555	135	311	66.10%	87.20%	50	74.30%	61.40%

Table 3. Results of ablation experiments on AIR-aircraft targets.

Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
Baseline	35	9	184	93.70%	94.80%	12	94.20%	89.30%
Baseline + Tracklets	41	9	182	93.70%	94.90%	5	94.10%	89.60%
Baseline + Transformer	31	5	183	95.10%	96.60%	5	95.90%	92.60%
Proposed Method	29	7	189	95.10%	96.50%	7	96.10%	92.70%

As can be seen from Table 3, in terms of MOTA, the proposed method was 3.30% higher than the baseline. For FM, ML, MT, IDR, IDP, IDs, and IDF1, the proposed method improved on the baseline by 6, 2, 5, 1.4%, 1.70%, 5, and 1.90%, respectively. This further

demonstrates the effectiveness of the proposed method on the AIR-aircraft dataset. On the one hand, transformer-based feature distillation increases the accuracy of object detection, which in turn improves multi-target tracking performance. On the other hand, the pixel-wise distillation method also improves performance.

4.6. Comparison to State-of-the-Art MOT Methods

4.6.1. Evaluation on the AIR-MOT Dataset

In order to evaluate the performance of the proposed method, we selected both DBT-based and JDT-based methods for comparison. Most of these methods are recently proposed, popular, and state-of-the-art methods. The DBT-based methods included StrongSORT [15], SORT [11], DeepSORT [13], GHOST [28], CKDNet-SMTNet [32], and DSFNet [33]. The JDT-based methods included FairMOT [21], TraDeS [22], SGT [50], UMA [17], CenterTrack [19], and TGraM [10]. In addition, we also selected transformer-based methods, including TrackFormer [30] and MOTR [38].

Table 4 shows the experimental results for aircraft and ship targets on the AIR-MOT dataset. From Table 4, we can see that the proposed method exceeded the second-ranked TGraM by 6.4% in terms of MOTA for aircraft targets. This demonstrates the superior performance of our method. For ship targets, the results showed that the proposed method exceeded the second-ranked method by 6.7% in terms of MOTA. The ship targets were usually small and closely arranged, which demonstrates that our method performs well on densely arranged, smaller targets. In addition, when the basic performance for aircraft targets was very high, our method still showed a significant improvement. This further verifies the superior performance of the proposed method.

4.6.2. Evaluation on the SAT-MTB Dataset

To verify the generalization performance of our method, we selected both DBT-based and JDT-based methods for comparison on another recently proposed remote sensing dataset, namely SAT-MTB [48]. The DBT-based methods included ByteTrack [12], DeepSORT [13], Trackor++ [51], CKDNet-SMTNet [32], DeepSORT [13], and Trackor++ [51]. The JDT-based methods included FairMOT [21] and TGraM [10].

Table 5 shows the experimental results for ship and airplane targets on the SAT-MTB dataset. In Table 5, it can be seen that the proposed method exceeded the second-ranked CenterTrack by 3.8% in terms of MOTA for ship targets. This demonstrates the state-of-the-art performance of the proposed method on the SAT-MTB database. From Table 5, we can also see that the proposed method exceeded the second-ranked method by 7.7% in terms of MOTA for airplane targets. Since ship targets were usually small and closely arranged, the results showed that the proposed method had the highest performance, which demonstrates that our method performs well on densely arranged, small targets. In addition, the basic tracking performance of airplane targets was already high, but the proposed method still exceeded the second-ranked method by 7.7% in terms of MOTA, further verifying the generalization performance of our method.

4.6.3. Evaluation on the VISO Dataset

We also verified the performance of our method on the VISO [31] dataset. The methods selected for comparison included MMB+FairMOT [31], DSFNet [33], MMB+CMOT [31], CKDNet-SMTNet [32], DSFNet+SORT [33], MMB+DTTP [31], and CenterTrack [19].

Table 6 shows the experimental results for car targets on the VISO dataset. From Table 6, it can be seen that the proposed method exceeded the second-ranked CenterTrack [19] by 6.1% in terms of MOTA. The proposed method also had the best performance in terms of IDF1 and IDR. This not only verifies the superior performance of our method on the VISO dataset but also verifies the generalization performance of our method.

Table 4. Comparison of results from the proposed method and several state-of-the-art methods on the AIR-MOT dataset.

AIR-aircraft								targets
Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
SORT [11]	16	17	183	90.0%	99.5%	7	94.5%	89.7%
UMA [17]	81	11	186	91.4%	92.8%	19	92.1%	85.4%
CenterTrack [19]	35	9	184	93.7%	94.8%	12	94.2%	89.3%
TraDeS [22]	114	44	148	75.0%	97.4%	42	84.7%	73.6%
FairMOT [21]	226	8	180	92.2%	94.2%	23	93.2%	86.7%
CKDNet-SMTNet [32]	52	3	187	95.7%	85.0%	36	90.0%	82.1%
TrackFormer [30]	110	12	184	90.5%	87.1%	97	88.8%	80.2%
TGraM [10]	181	8	174	90.7%	94.1%	28	92.4%	86.3%
StrongSORT [15]	14	18	182	88.2%	99.6%	1	93.5%	87.8%
SGT [50]	153	6	184	93.9%	94.6%	56	94.3%	89.2%
GHOST [28]	67	15	183	90.1%	98.8%	2	94.3%	89.6%
DSFNet-SORT [33]	96	28	164	83.1%	98.9%	6	90.3%	82.3%
Proposed Method	29	7	189	95.1%	96.5%	7	96.1%	92.7%
AIR-ship targets								
Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
SORT [11]	434	254	236	49.7%	92.0%	80	64.5%	47.5%
CenterTrack [19]	640	143	297	52.4%	70.3%	2461	60.1%	54.4%
UMA [17]	1193	204	277	56.2%	71.6%	250	62.9%	39.8%
CKDNet-SMTNet [32]	1375	238	175	40.1%	59.5%	516	47.9%	25.0%
FairMOT [21]	2487	262	187	43.3%	77.4%	258	55.5%	34.7%
TraDeS [22]	1265	261	202	47.9%	75.7%	118	58.6%	35.0%
TGraM [10]	1449	311	124	31.7%	71.7%	166	43.9%	25.1%
TrackFormer [30]	1439	173	286	55.6%	62.7%	1360	58.9%	34.9%
StrongSORT [15]	311	312	206	42.4%	95.9%	19	58.8%	41.0%
SGT [50]	1718	238	217	47.4%	87.7%	717	61.5%	44.8%
DSFNet-SORT [33]	1045	352	135	33.2%	83.3%	36	47.5%	27.0%
GHOST [28]	927	246	239	50.5%	90.8%	162	64.9%	47.8%
Proposed Method	555	135	311	66.1%	87.2%	50	74.3%	61.4%

Table 5. Comparison of results from the proposed method and several state-of-the-art methods on the SAT-MTB dataset.

SAT-MTB-ship targets								
Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
DeepSORT [13]	-	40	35	-	-	144	45.5%	13.5%
Trackor++ [51]	-	30	35	-	-	119	56.6%	34.3%
CenterTrack [19]	325	13	39	68.5%	69.4%	84	68.3%	54.1%
FairMOT [21]	-	39	23	-	-	143	45.0%	8.2%
CKDNet-SMTNet [32]	-	14	43	-	-	299	49.8%	36.9%
TGraM [10]	-	47	19	-	-	283	35.6%	−13.9%
ByteTrack [12]	-	34	38	-	-	168	46.9%	10.5%
DSFNet [33]	-	16	38	-	-	235	57.7%	37.2%
Proposed Method	313	11	45	69.2%	72.0%	82	71.6%	57.9%
SAT-MTB-airplane targets								
Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
DeepSORT [13]	-	33	22	-	-	0	55.3%	35.6%
Trackor++ [51]	-	13	33	-	-	64	57.8%	17.6%
CenterTrack [19]	38	8	45	76.2%	77.7%	38	76.9%	56.7%
FairMOT [21]	-	22	32	-	-	1	63.9%	32.0%
CKDNet-SMTNet [32]	-	3	53	-	-	91	65.9%	54.3%
TGraM [10]	-	18	33	-	-	6	67.5%	37.4%
ByteTrack [12]	-	26	31	-	-	1	65.1%	45.3%
Proposed Method	36	7	47	74.1%	82.3%	66	77.0%	64.4%

Table 6. Comparison of results from the proposed method and several state-of-the-art methods on the VISO dataset (car targets).

Method	FM	ML	MT	IDR	IDP	IDs	IDF1	MOTA
CenterTrack [19]	3086	328	275	34.6%	67.3%	5503	45.7%	45.3%
CKDNet-SMTNet [32]	2887	443	105	22.6%	62.8%	3071	33.2%	33.5%
MMB+FairMOT [31]	205	623	21	-	-	52	-	2.3%
DSFNet [33]	1326	432	45	-	-	348	-	3.6%
MMB+CMOT [31]	111	494	38	-	-	89	-	22.8%
DSFNet+SORT [33]	8075	105	318	-	-	434	-	41.6%
MMB+DTTP [31]	1344	22	483	-	-	3090	-	44.5%
Proposed Method	3019	277	355	40.1%	67.4%	5810	49.3%	51.4%

In addition to quantitative comparisons, we also visualized experimental results on several sequences, as shown in Figure 6. To save space, we only present visualizations of the top three methods. As can be seen in Figure 6, our method performed the best in terms of ID continuity, missed detections, and false detections. This also demonstrates that the use of pixel-wise transformer distillation and temperature-controllable key feature learning is obviously superior to existing methods for remote sensing targets.

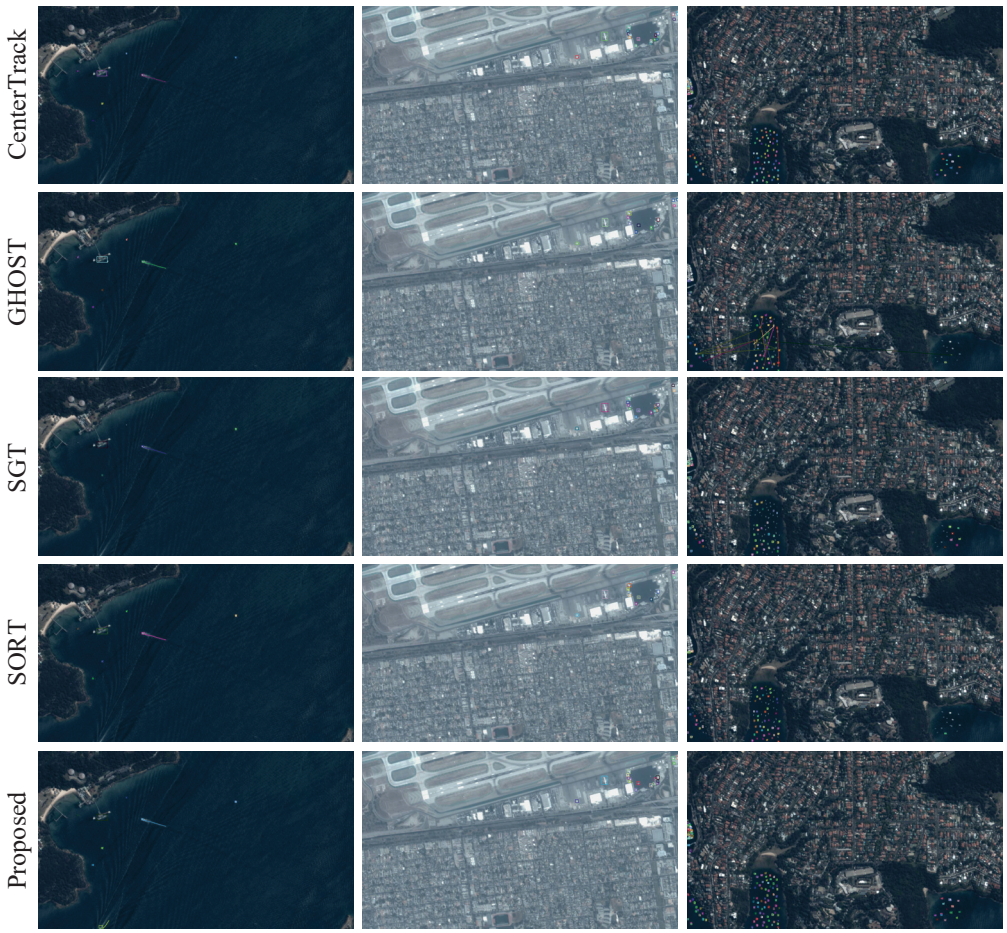


Figure 6. *Cont*.

Figure 6. Visualization of several tracking results using the baseline method (CenterTrack) and the proposed method.

5. Conclusions

MOT in satellite videos is significant in many important applications. Due to the limitations of satellite computing devices and the need for high tracking performance for many crucial applications, designing efficient and accurate tracking frameworks is imperative. As a consequence, in this paper, we propose a new transformer distillation-based end-to-end joint detection and tracking method. In this method, using a highly precise teacher network, object features are distilled into a JDT method to improve localization and tracking accuracy while maintaining tracking efficiency. Considering the scale variance between different objects, such as ships and aircraft, we propose a pixel-wise transformer distillation module to effectively distill useful target features. Moreover, a temperature-controllable feature distillation objective function is also proposed to distill critical features. Experiments on aircraft, ship, and car targets demonstrated the significant improvements offered by the proposed method over the baseline and current state-of-the-art methods. The proposed method achieved an overall tracking performance of over 90% on the AIR-MOT dataset. The proposed method inherits advantages from both DBT- and JDT-based methods during tracking. However, it reduces training efficiency with object detection network and distillation learning. In future work, we will try to use existing large remote sensing models to facilitate the training of the tracking network.

Author Contributions: Z.S.: Methodology, Software, and Writing—original draft; G.W. (Gang Wan), Y.W., N.G. and D.C.: Supervision; Z.W., W.L. and G.W. (Guoping Wang): Validation and Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Internal Parenting Program (Grant No.: 145AXL250004000X) and the State Key Laboratory of Geo-Information Engineering (Grant No.: SKLGIE2022-ZZ2-08).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are unavailable due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, J.; Hua, W.; Zhang, W.; Liu, F.; Xiao, L. Stair Fusion Network With Context-Refined Attention for Remote Sensing Image Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4701517. [CrossRef]
2. Zhang, J.; Jia, X.; Hu, J.; Chanussot, J. Online Structured Sparsity-Based Moving-Object Detection From Satellite Videos. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6420–6433. [CrossRef]
3. Zhang, W.; Jiao, L.; Liu, F.; Yang, S.; Liu, J. Adaptive Contourlet Fusion Clustering for SAR Image Change Detection. *IEEE Trans. Image Process.* **2022**, *31*, 2295–2308. [CrossRef] [PubMed]
4. Zhang, W.; Jiao, L.; Liu, F.; Yang, S.; Song, W.; Liu, J. Sparse Feature Clustering Network for Unsupervised SAR Image Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5226713. [CrossRef]
5. Zhang, W.; Jiao, L.; Li, Y.; Huang, Z.; Wang, H. Laplacian Feature Pyramid Network for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5604114. [CrossRef]
6. Zhang, W.; Jiao, L.; Liu, F.; Liu, J.; Cui, Z. LHNet: Laplacian Convolutional Block for Remote Sensing Image Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5626513. [CrossRef]
7. Chen, H.; Li, N.; Li, D.; Lv, J.; Zhao, W.; Zhang, R.; Xu, J. Multiple Object Tracking in Satellite Video With Graph-Based Multi-Clue Fusion Tracker. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5639914. [CrossRef]
8. Zhang, W.; Jiao, L.; Liu, F.; Li, L.; Liu, X.; Liu, J. MBLT: Learning Motion and Background for Vehicle Tracking in Satellite Videos. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4703315. [CrossRef]
9. Wang, B.; Ma, G.; Sui, H.; Zhou, Y.; Zhang, H.; Liu, J. Multi-Object Tracking in Satellite Videos Considering Weak Feature Enhancement. In Proceedings of the IGARSS 2024—2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; pp. 8082–8085.
10. He, Q.; Sun, X.; Yan, Z.; Li, B.; Fu, K. Multi-Object Tracking in Satellite Videos With Graph-Based Multitask Modeling. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5619513. [CrossRef]
11. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA , 25–28 September 2016; pp. 3464–3468. [CrossRef]
12. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
13. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649. [CrossRef]
14. Sun, S.; Akhtar, N.; Song, H.; Mian, A.; Shah, M. Deep Affinity Network for Multiple Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 104–119. [CrossRef] [PubMed]
15. Du, Y.; Song, Y.; Yang, B.; Zhao, Y. StrongSORT: Make DeepSORT Great Again. *arXiv* **2022**, arXiv:2202.13514. [CrossRef]
16. Bochinski, E.; Eiselein, V.; Sikora, T. High-Speed tracking-by-detection without using image information. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6. [CrossRef]
17. Yin, J.; Wang, W.; Meng, Q.; Yang, R.; Shen, J. A Unified Object Motion and Affinity Model for Online Multi-Object Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6767–6776. [CrossRef]
18. Dai, P.; Weng, R.; Choi, W.; Zhang, C.; He, Z.; Ding, W. Learning a Proposal Classifier for Multiple Object Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2443–2452. [CrossRef]
19. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking Objects as Points. In *Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part IV*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12349, pp. 474–490.
20. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. In *Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XI*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12356, pp. 107–122.
21. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
22. Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; Yuan, J. Track To Detect and Segment: An Online Multi-Object Tracker. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; pp. 12352–12361.

23. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
24. Yang, F.; Choi, W.; Lin, Y. Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2129–2137. [CrossRef]
25. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [CrossRef]
26. Basar, T. A New Approach to Linear Filtering and Prediction Problems. In *Control Theory: Twenty-Five Seminal Papers (1932–1981)*; Wiley-IEEE Press: Hoboken, NJ, USA, 2001; pp. 167–179. [CrossRef]
27. Kuhn, H.W. The Hungarian Method for the Assignment Problem. In *50 Years of Integer Programming 1958–2008—From the Early Years to the State-of-the-Art*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 29–47.
28. Seidenschwarz, J.; Brasó, G.; Serrano, V.C.; Elezi, I.; Leal-Taixé, L. Simple Cues Lead to a Strong Multi-Object Tracker. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 13813–13823. [CrossRef]
29. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
30. Meinhardt, T.; Kirillov, A.; Leal-Taixé, L.; Feichtenhofer, C. TrackFormer: Multi-Object Tracking with Transformers. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8834–8844. [CrossRef]
31. Yin, Q.; Hu, Q.; Liu, H.; Zhang, F.; Wang, Y.; Lin, Z.; An, W.; Guo, Y. Detecting and Tracking Small and Dense Moving Objects in Satellite Videos: A Benchmark. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5612518. [CrossRef]
32. Feng, J.; Zeng, D.; Jia, X.; Zhang, X.; Li, J.; Liang, Y.; Jiao, L. Cross-frame keypoint-based and spatial motion information-guided networks for moving vehicle detection and tracking in satellite videos. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 116–130. [CrossRef]
33. Zhang, J.; Zhang, X.; Huang, Z.; Cheng, X.; Feng, J.; Jiao, L. Bidirectional Multiple Object Tracking Based on Trajectory Criteria in Satellite Videos. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5603714. [CrossRef]
34. Wu, J.; Su, X.; Yuan, Q.; Shen, H.; Zhang, L. Multivehicle Object Tracking in Satellite Video Enhanced by Slow Features and Motion Features. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5616426. [CrossRef]
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; 2017 ; pp. 5998–6008.
36. Zhou, X.; Yin, T.; Koltun, V.; Krähenbühl, P. Global Tracking Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8761–8770.
37. Cai, J.; Xu, M.; Li, W.; Xiong, Y.; Xia, W.; Tu, Z.; Soatto, S. MeMOT: Multi-Object Tracking with Memory. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8080–8090.
38. Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; Wei, Y. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *Proceedings of the Computer Vision—ECCV 2022—17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XXVII*; Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13687, pp. 659–675.
39. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part I*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12346, pp. 213–229.
40. Zhang, Y.; Wang, T.; Zhang, X. MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors. *arXiv* **2022**, arXiv:2211.09791.
41. Zhao, Z.; Wu, Z.; Zhuang, Y.; Li, B.; Jia, J. Tracking Objects as Pixel-Wise Distributions. In *Proceedings of the Computer Vision—ECCV 2022—17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XXII*; Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13682, pp. 76–94.
42. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
43. Huang, Y.; Wu, J.; Xu, X.; Ding, S. Evaluation-oriented Knowledge Distillation for Deep Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 18719–18728. [CrossRef]
44. Li, Z.; Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]

45. Ma, H.; Li, J.; Hosseini, R.; Tomizuka, M.; Choi, C. Multi-Objective Diverse Human Motion Prediction with Knowledge Distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8151–8161. [CrossRef]
46. Chen, Q.; Zhong, B.; Liang, Q.; Deng, Q.; Li, X. Teacher-student knowledge distillation for real-time correlation tracking. *Neurocomputing* **2022**, *500*, 537–546. [CrossRef]
47. Zhang, W.; He, L.; Chen, P.; Liao, X.; Liu, W.; Li, Q.; Sun, Z. Boosting End-to-end Multi-Object Tracking and Person Search via Knowledge Distillation. In Proceedings of the MM'21: ACM Multimedia Conference, Virtual Event, China, 20–24 October 2021; Shen, H.T., Zhuang, Y., Smith, J.R., Yang, Y., César, P., Metze, F., Prabhakaran, B., Eds., ACM: New York, NY, USA, 2021; pp. 1192–1201. [CrossRef]
48. Li, S.; Zhou, Z.; Zhao, M.; Yang, J.; Guo, W.; Lv, Y.; Kou, L.; Wang, H.; Gu, Y. A Multitask Benchmark Dataset for Satellite Video: Object Detection, Tracking, and Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5611021. [CrossRef]
49. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
50. Hyun, J.; Kang, M.; Wee, D.; Yeung, D.Y. Detection Recovery in Online Multi-Object Tracking with Sparse Graph Tracker. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 4839–4848. [CrossRef]
51. Bergmann, P.; Meinhardt, T.; Leal-Taixé, L. Tracking Without Bells and Whistles. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 941–951. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Neural Radiance Fields for Fisheye Driving Scenes Using Edge-Aware Integrated Depth Supervision

Jiho Choi and Sang Jun Lee *

Division of Electronics and Information Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea; jihochoi@jbnu.ac.kr
* Correspondence: sj.lee@jbnu.ac.kr

Abstract: Neural radiance fields (NeRF) have become an effective method for encoding scenes into neural representations, allowing for the synthesis of photorealistic views of unseen views from given input images. However, the applicability of traditional NeRF is significantly limited by its assumption that images are captured for object-centric scenes with a pinhole camera. Expanding these boundaries, we focus on driving scenarios using a fisheye camera, which offers the advantage of capturing visual information from a wide field of view. To address the challenges due to the unbounded and distorted characteristics of fisheye images, we propose an edge-aware integration loss function. This approach leverages sparse LiDAR projections and dense depth maps estimated from a learning-based depth model. The proposed algorithm assigns larger weights to neighboring points that have depth values similar to the sensor data. Experiments were conducted on the KITTI-360 and JBNU-Depth360 datasets, which are public and real-world datasets of driving scenarios using fisheye cameras. Experimental results demonstrated that the proposed method is effective in synthesizing novel view images, outperforming existing approaches.

Keywords: neural radiance field; view synthesis; depth supervision; fisheye camera

1. Introduction

Neural rendering with coordinate-based and implicit functions has evolved as a method for representing 3D objects and scenes from 2D images. Neural volumetric representations have been utilized in various computer vision problems, including novel view synthesis [1], relighting [2], editing [3,4] and pose estimation [5,6], as well as various industrial applications [7,8]. In particular, neural radiance fields (NeRF) [1] have successfully rendered from novel viewpoints with implicit features, resulting in photorealistic image synthesis. NeRF enables us to capture scene geometry with view-dependent emitted radiance with volume density.

NeRF has primarily focused on object-centric scenarios and has conducted experiments using data that specifically revolve around individual objects. Object-centric datasets assume a single, isolated object in a scene, which can lead to challenges when rendering scenes with multiple or complex objects. Moreover, real-world scenes rarely have a single object at the center, highlighting the need for diverse camera trajectories to ensure the scalability of the NeRF. Urban-NeRF [9] aims to render urban environments and enables 3D reconstruction of complex scenes with multiple objects such as buildings. Similarly, S-NeRF [10] extends the applicability of NeRF to realistic and dynamic driving scenarios by proposing street view synthesis using driving datasets. These approaches show ongoing efforts to overcome the limitations of object-centric NeRF algorithms and expand their scope to encompass more diverse and complex scenes.

Previous NeRF algorithms operate predominantly under the assumption of a pinhole camera model. The pinhole model assumes the transmission of light from the real world through a pinhole, forming an inversely oriented image on the camera sensor. However,

this approach imposes limitations, primarily the restrictive field of view (FoV) of a pinhole camera, which significantly reduces the capacity to capture comprehensive scenes in a single frame. Furthermore, the idealized pinhole model does not account for real-world camera properties, such as lens distortion. Consequently, NeRF algorithms founded on this model may not provide universally applicable solutions. In response to these limitations, the research introduces a fisheye camera model within the NeRF algorithm. This approach broadens the applicability and enhances the efficiency of scene capture.

Compared to pinhole cameras, fisheye cameras offer the advantage of a wider FoV, typically ranging from 180° to 220°. A wide FoV camera can capture over 180° with a single lens and is economically efficient, as it reduces the number of cameras needed to cover an omnidirectional scene. Therefore, fisheye cameras are widely used in autonomous vehicles such as surround monitoring and scene understanding. Previously, various approaches have been explored to utilize fisheye cameras by undistorting the images to address their significant distortion. However, this process results in a loss of image boundary information and degrades overall image quality. To fully exploit the wide FoV, recent efforts have focused on using original fisheye images, and large-scale driving datasets captured by fisheye cameras have been introduced [11–13]. In light of this, the proposed method utilizes the original fisheye images without applying distortion correction.

In this study, we propose an edge-aware integration loss function designed to minimize the difference in the distribution of rendered rays and integrated depth. The proposed method utilizes accurate depth data obtained from the LiDAR sensor, coupled with estimated dense depth information, which carries inherent uncertainty. Specifically, pixels adjacent to the LiDAR point rely more on sensor data than on the estimated depth. Moreover, we consider the edge component by assigning larger weights to pixels that have similar depth. Figure 1 shows an overview of the proposed method. The effectiveness of the proposed algorithm was evaluated on driving datasets including the KITTI-360 and the custom dataset.

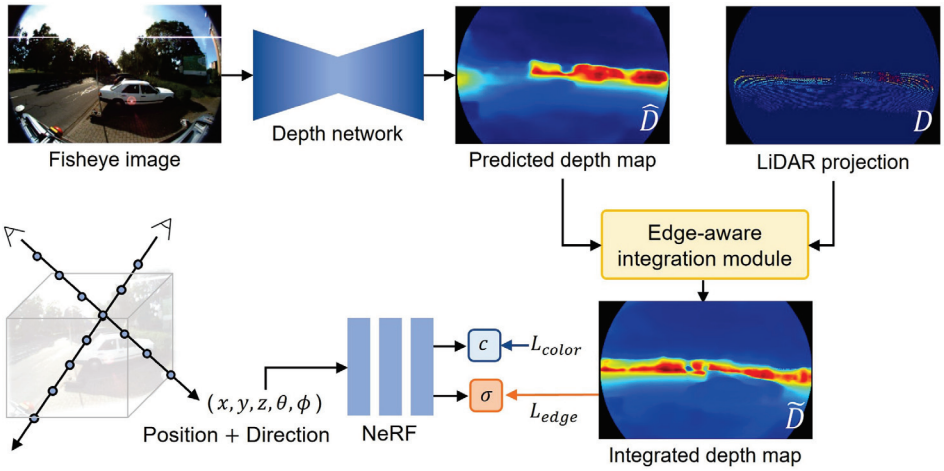


Figure 1. Overview of the proposed pipeline. Given a set of RGB images captured by fisheye camera in driving scenarios, we trained a monocular depth estimation network that outputs the densely predicted depth map $\hat{D}$. Moreover, the LiDAR points are projected to generate a sparse depth map D. These two depth priors are employed in the edge-aware integration module. The training of the radiance field is guided by the RGB images and the integrated depth maps $\tilde{D}$, which inform the model regarding ray termination. NeRF takes 5D inputs and is trained using the L_{color} and the proposed L_{edge} loss functions.

In summary, we propose a NeRF that employs a fisheye camera model aimed at driving scenarios, rather than a conventional object-centric approach. The implementation code is available at https://github.com/ziiho08/Edge-aware-Integrated-Depth-Supervision (accessed on 17 October 2024). The contributions are the following:

- We present a method that utilizes a fisheye camera to synthesize novel views in real-world driving scenarios.
- We propose an edge-aware integration loss function that minimizes the difference between the rendered ray and the integrated depth distribution.
- We demonstrate that the proposed method outperforms other approaches, as shown by the results from the KITTI-360 [13] and JBNU-Depth360 datasets.

2. Related Work

2.1. Fisheye and Omni-Directional NeRF

NeRF contains the radiance field of a scene in a neural network and generates photo-realistic renderings of novel views. Various approaches have been proposed to improve rendering performance in terms of anti-aliasing [14], faster training and inference [15], and generalization [16]. However, most previous methods rely on the pinhole camera model, and research on other camera models remains limited. Recent works have proposed NeRF based on camera models other than the traditional pinhole camera. Omni-NeRF [17] proposed a spherical sampling method for synthesizing novel views using fisheye images, demonstrating scene reconstruction for both synthetic and real fisheye camera datasets. 360FusionNeRF [18] introduced methods to render novel views from 360° panoramic images in RGB-D, using geometric supervision to enhance rendering accuracy. Moreover, DiCo-NeRF [19] integrated a vision-language model to enhance the performance of novel view synthesis in fisheye driving scenes. These studies demonstrate that NeRF is not limited to a specific camera type but can be effectively applied to various camera models, including fisheye cameras. Notably, the wide FoV camera enable the rendering of broad scenes within a single image.

The SCNeRF [6] approach proposes a method for handling generic camera models by performing self-calibration to optimize NeRF without relying on pose information. This technique enables the rendering of generic camera models, including those with complex non-linear camera distortion coefficients. The study showed experimental results on the fisheye image dataset and demonstrated its applicability to various variants of NeRF. However, they predominantly focus on static fisheye images, suggesting the potential for further exploration in dynamic scenarios. To address this gap, our work presents a method for synthesizing novel views in driving scenarios using fisheye images, thereby expanding the applicability of NeRF algorithms.

2.2. NeRF with Depth Supervision

Several studies have successfully performed 3D reconstruction by supervising the NeRF model with additional depth information. NerfingMVS [20] introduces an optimization framework that incorporates adapted learning-based depth priors to regulate the sampling process of NeRF during volume rendering. The experimental results of the indoor ScanNet dataset demonstrate the effectiveness of their approach. Moreover, DS-NeRF [21] leverages sparse 3D point clouds from structure-from-motion (SFM) COLMAP [22] to improve rendering results with fewer training views, while achieving faster training speeds compared to traditional NeRF. Several studies demonstrated promising results in utilizing LiDAR data for novel view synthesis tasks. Urban NeRF [9] utilizes RGB images and asynchronously recorded LiDAR points to enhance the reconstruction of large outdoor scenes. They have demonstrated state-of-the-art 3D surface reconstructions and realistic novel views in their experiments conducted on street view data. S-NeRF [10] takes advantage of sparse LiDAR data from autonomous vehicles to enhance neural representation learning, improving street view synthesis. They simultaneously rendered large-scale background scenes and moving foreground vehicles, and conducted experiments on substantial driving datasets, namely nuScenes [23] and Waymo [24].

Recently, SparseNeRF [25] introduced local depth ranking to enhance performance with limited views by utilizing imprecise depth priors. Additionally, [26] provided a comprehensive review of the use of depth priors in outdoor scenarios for NeRF, evaluating

various depth types to evaluate their influence on performance. These methodologies have shown promising results on large driving datasets, underlining their ability to effectively reconstruct urban and street scenes.

3. Method

We propose a novel NeRF algorithm that utilizes depth supervision. Synthesizing wide FoV images in driving scenarios is a difficult problem due to the strong distortion and unbounded characteristics. To address these challenges, we introduced dense depth maps inferred by a learning-based model to supplement sparse LiDAR projections. The edge-aware integration loss is a function for effectively employing the sparse and dense depth maps.

3.1. Preliminaries

A NeRF encodes scenes by representing them as a color value $\mathbf{c} \in [0,1]^3$ and density $\sigma \in [0,\infty)$. This representation is achieved through a multi-layer perceptron (MLP), which serves as an implicit function for modeling the scenes. The NeRF model receives a 3D coordinate $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{S}^2$ as inputs and predicts the corresponding density and RGB color values. This process is expressed by the equation: $f(\mathbf{x}, \mathbf{d}) = (\sigma, \mathbf{c})$, where f is the MLP that generates the density and RGB color values. To capture high-frequency details, NeRF applies positional encoding to its input, mapping them to a higher-dimensional space using sinusoidal functions of various frequencies. This encoding process is defined as follows:

$$\gamma_L(\mathbf{x}) = \Big[sin(\mathbf{x}), cos(\mathbf{x}), ..., sin(2^{L-1}\mathbf{x}), cos(2^{L-1}\mathbf{x})\Big], \tag{1}$$

where L is a parameter that controls the maximum frequency used in the positional encoding [1]. For rendering a pixel's intensity, a ray $\mathbf{r}(t)$ is cast from the camera's center in the direction $\mathbf{d}$, with t representing the distance. The color value is obtained by integrating the radiance fields from t_n to t_f as described by the following Equation [1]:

$$\begin{aligned} \hat{\mathbf{c}}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \\ where \quad T(t) &= exp(-\int_{t_n}^{t_f} \sigma(\mathbf{r}(s))ds). \end{aligned} \tag{2}$$

In Equation (2), σ and $\mathbf{c}$ are the density and color values predicted by the function f, and $T(t)$ represents the transmittance along the ray up to the distance t. The ray termination distribution is denoted as $h(t)$ and can be calculated as $h(t) = T(t)\sigma(t)$. The NeRF model is trained by minimizing the color difference between the predicted and actual values [1], as denoted in the loss function:

$$L_{color} = \sum_{\mathbf{r} \in R_i} \|\hat{\mathbf{c}}(\mathbf{r}) - \mathbf{c}(\mathbf{r})\|^2, \tag{3}$$

where R_i represents the set of rays for the i-th image.

3.2. Depth Supervision

The proposed method employs sparse LiDAR projections and dense depth predictions. The LiDAR projections are denoted and $D = \left\{D_i \in \mathbb{R}^{H \times W} \mid i = 0, \ldots, N-1\right\}$, where H and W represent the height and width, and N is the number of images. Although the LiDAR projection provides accurate depth information, it suffers from a sparsity of valid points. To address this limitation, we incorporate depth predictions, denoted as $\hat{D} = \left\{\hat{D}_i \in \mathbb{R}^{H \times W} \mid i = 0, \ldots, N-1\right\}$, inferred from a learning-based depth network. To simplify the equation, we will omit the subscript i in the following explanations.

Recent advancements in supervised depth estimation algorithms have shown promising results in multiple datasets including NYU-Depth-v2 [27] and KITTI [28]. Motivated by the success, we aim to improve the performance of a NeRF model by incorporating depth information inferred from a depth estimation model. We trained a monocular depth estimation model [29], which takes an image to estimate a dense depth map $\hat{D}$. While the predicted depth map contains the knowledge of terminating the rays, it contains uncertainty for predicting the accurate depth information. To address the sparsity of the D and uncertainty of $\hat{D}$, we integrated the sparse and dense depth maps to improve the reliability of the depth prior.

3.3. Edge-Aware Integration Loss

In autonomous driving scenarios using fisheye images, novel view synthesis presents unique challenges due to unbounded and distorted characteristics. To address these, we leverage the depth information derived from LiDAR points in the driving datasets as shown in Figure 2. This accurate depth guides the NeRF model in terms of ray termination. Our objective is to minimize the difference between the rendered ray distribution h_k and integrated depth $\tilde{d}_{i,j}$, and it is referred to as the edge-aware integration loss:

$$L_{edge} = \sum_{i,j} \sum_{k} log\, h_k\, exp(-\frac{(t_k - \tilde{d}_{i,j})^2}{2\hat{\sigma}^2})\Delta t_k, \tag{4}$$

where h_k denotes the termination distribution of the ray, Δt_k represents the ray step, and $\hat{\sigma}$ is the standard deviation of integrated depth map $\tilde{D}$. The components of $\tilde{D}$ at the pixel (i, j) is indicated as $\tilde{d}_{i,j}$, and it can be computed as

$$\begin{aligned} \tilde{d}_{i,j} = \eta_{i,j}^{-1} \sum_{k,l} g_{i,j}^{(k,l)}(\hat{D}) \Big[& \alpha \cdot d_{i+k,j+l} \cdot 1(d_{i+k,j+l} > 0) \\ & + (1-\alpha) \cdot \hat{d}_{i+k,j+l} \cdot 1(d_{i+k,j+l} = 0) \Big]. \end{aligned} \tag{5}$$

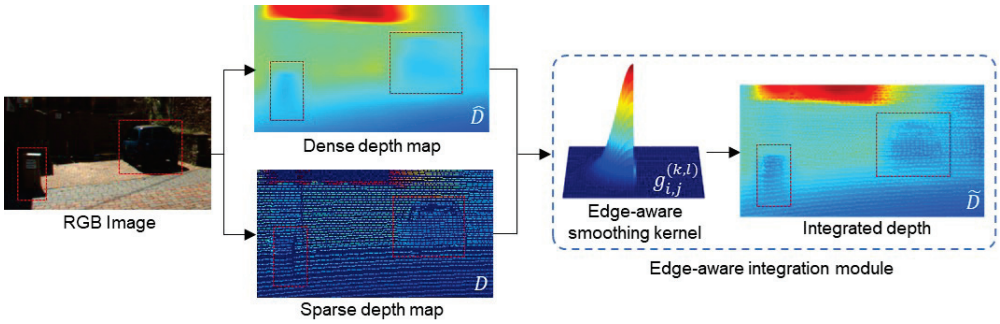


Figure 2. Process of integrating the sparse LiDAR projection and dense estimated depth map. We propose the edge-aware integration loss function to optimize the NeRF model with depth supervision, as detailed in (4). The proposed method guides the NeRF model using depth priors from the scene by minimizing the difference between the distributions of ray termination from the model and the given depth information. This depth is determined using the edge-aware smoothing kernel, which takes advantage of both depth priors. Moreover, it assigns a larger weight to adjacent points that are consistent with depth values.

We obtain $\tilde{d}_{i,j}$ by integrating a sparse LiDAR projection map with a dense prediction map using the explicit function, and η_{ij} is the normalization factor defined as

$$\begin{aligned} \eta_{i,j} = \sum_{k,l} g_{i,j}^{(k,l)}(\hat{D}) \Big[& \alpha \cdot 1(d_{i+k,j+l} > 0) \\ & + (1-\alpha) \cdot 1(d_{i+k,j+l} = 0) \Big], \end{aligned} \tag{6}$$

where α is a hyper-parameter that controls the effect of sparse and dense depth maps, and it was set to 0.7. In experiments, the kernel size was set to 9, and the indexes k and l are integers in $[-4, 4]$.

In the edge-aware integration module, as described in (5), we factor in the locations of the neighbors of the pixel (i, j) within the kernel to integrate two distinct depth maps. The dense depth $\hat{d}_{i+k,j+l}$ surrounding the LiDAR point $d_{i,j}$ is refined using neighboring sensor data. This refinement involves assigning a larger weight to pixels closer to the LiDAR information. Importantly, pixel values outside an object's edge differ significantly from the depth values within its boundary. To address this, we consider the depth differences between LiDAR and inferred depth to resolve the depth ambiguity. We term these procedures the edge-aware smoothing kernel, denoted as

$$g_{i,j}^{(k,l)}(\hat{D}) = exp(-\frac{k^2 + l^2}{2\sigma_s^2}) \cdot exp(-\frac{(\hat{d}_{i,j} - \hat{d}_{i+k,j+l})^2}{2\sigma_d^2}), \tag{7}$$

where σ_s and σ_d represent the standard deviation of spatial and depth spaces; they were set to 3 and 1, respectively. This smoothing kernel helps to preserve edge components while ensuring the diffusion of accurate depth data from the sensor.

The proposed approach aims to optimize the NeRF for fisheye driving scenarios. We achieve this by employing the proposed edge-aware integration loss function, which leverages two different depth priors. The total loss function for NeRF is $L_{total} = L_{color} + \lambda L_{edge}$, and λ determines the balance between the loss functions.

4. Experimental Results

We evaluated the proposed method using the KITTI-360 [13] and JBNU-Depth360 datasets, which include fisheye driving scenes in both outdoor and parking lot environments. In these experiments, we compared the method with previous NeRF algorithms, including nerfacto [30], UrbanNeRF [9], and DS-NeRF [21]. In addition, ablation studies were conducted to demonstrate the effectiveness of the edge-aware integration module and to explore the impact of different depth types, including the LiDAR projection and the estimated depth from a depth network.

4.1. Datasets

- KITTI-360.

This dataset includes 320 k images and 100 k laser scans collected from driving scenarios in the suburbs of Karlsruhe, Germany. It contains RGB images captured with 180° FoV fisheye cameras and 3D point clouds acquired from an HDL-64E Velodyne LiDAR sensor. The fisheye images have a resolution of 1400 × 1400, and the point clouds were projected onto the 2D image plane to generate sparse ground-truth depth maps. The dataset encompasses nine different driving scenarios and we used 150 images from each scenario to train the NeRF model. In the experiments, we utilized known intrinsic parameters, and camera poses were determined using the COLMAP algorithm.

- JBNU-Depth360.

This custom dataset was collected in underground parking lot environments using a Jackal UGV mobile robot equipped with a NVIDIA Jetson AGX Xavier, 183° FoV fisheye cameras, and a 90° FoV Ouster LiDAR sensor. The dataset includes 4221 RGB images with a resolution of 1080 × 1920, and the 3D LiDAR points were projected onto the image plane to generate sparse ground-truth depth maps. It consists of six driving scenarios, and we utilized approximately 126 images for the training of a NeRF model in each scenario. The training images were selected based on time intervals, and the relative camera poses were computed by using the COLMAP.

- Evaluation metric.

We report the quantitative comparison results using metrics such as the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS) in the output synthesized images of novel views.

4.2. Results on the KITTI-360 Dataset

We compare the method with nerfacto [30], UrbanNeRF [9] and DS-NeRF [21] on the KITTI-360 datasets. Table 1 shows the image synthesis metrics achieved by the proposed method on KITTI-360 dataset. We compare the proposed method with the base pipeline nerfacto, as well as the loss functions of DS-NeRF and UrbanNeRF. We tested these previous depth-based methods using the LiDAR projection as the ground truth. In most driving scenarios, our approach outperforms the previous depth supervision methods across all metrics. Specifically, the proposed method achieves average PSNR, SSIM, and LPIPS of 12.427, 0.535, and 0.574, respectively. The findings indicate that optimizing the NeRF model with depth supervision improves geometry accuracy by better understanding the ray termination distribution. This optimization is effective in reproducing the photorealistic fisheye images.

Table 1. Quantitative results on the KITTI-360 dataset. Higher is better for PSNR and SSIM and lower is better for LPIPS.

Method	PSNR↑				SSIM↑				LPIPS↓			
	Drive_0002	Drive_0007	Drive_0009	Average	Drive_0002	Drive_0007	Drive_0009	Average	Drive_0002	Drive_0007	Drive_0009	Average
Nerfacto [30]	13.466	13.699	13.650	13.024	0.580	0.583	0.627	0.537	0.532	0.607	0.439	0.583
UrbanNeRF [9]	13.645	13.765	13.997	13.099	0.579	0.580	0.629	0.539	0.551	0.603	0.436	0.590
DS-NeRF [21]	13.673	14.262	14.023	13.240	**0.580**	0.577	0.631	0.538	0.532	0.603	0.439	0.584
Ours	**14.180**	**14.920**	**14.102**	**13.678**	0.577	**0.594**	**0.641**	**0.548**	**0.521**	**0.594**	**0.641**	**0.557**

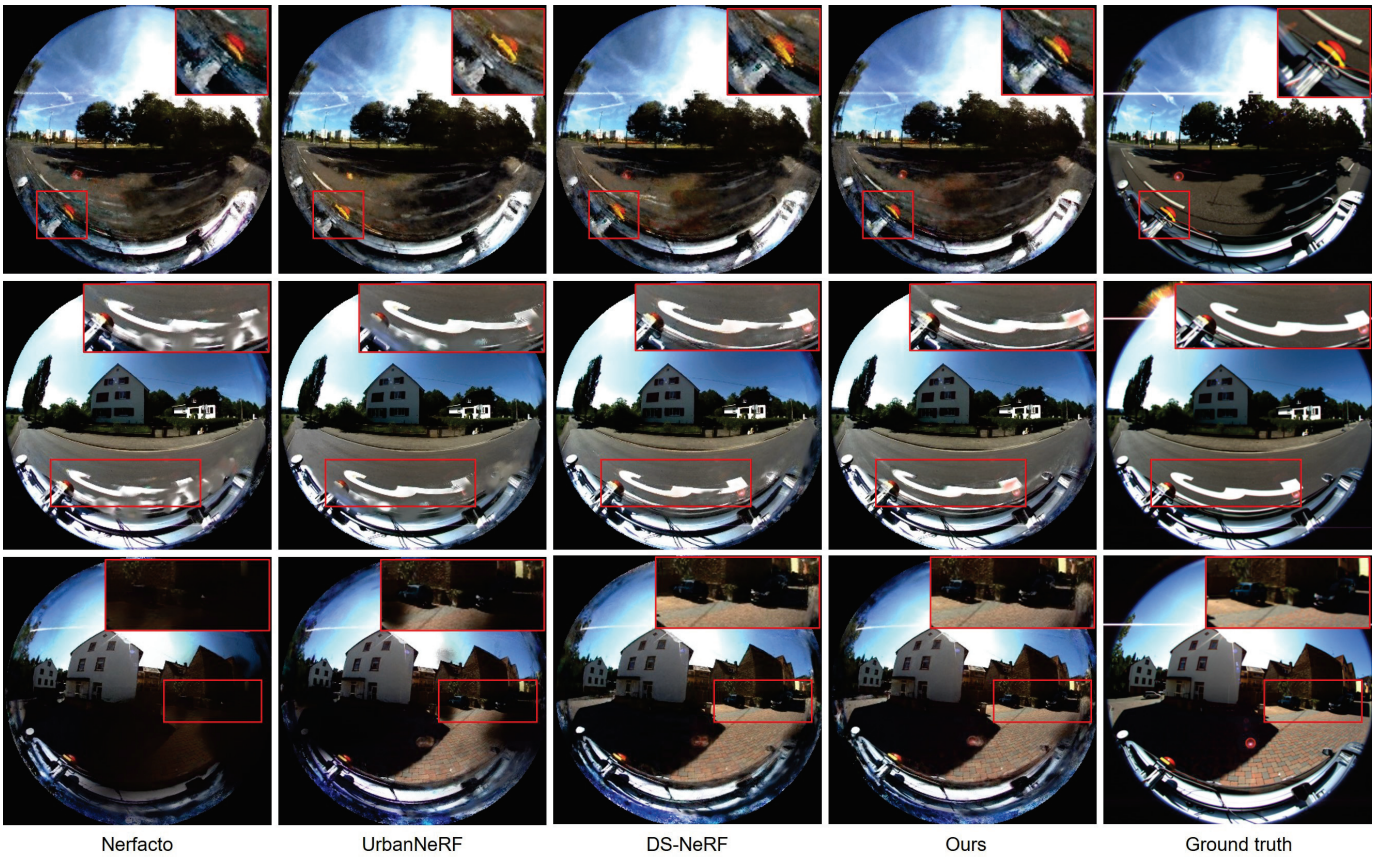


Figure 3. View synthesis on KITTI-360 dataset. The proposed method has demonstrated improved photorealistic results, as highlighted in the red boxes.

Figure 3 further highlights the effectiveness of the proposed approach through qualitative results. As shown in the first row, the proposed method's radiance representation avoids the blurry and floating artifacts commonly observed in other methods, particularly for objects near the camera. Additionally, it consistently produces sharper results for devices mounted on the vehicle. By optimizing NeRF with depth priors, the approach effectively reduces these artifacts, resulting in more realistic color synthesis and appearance. The results clearly depict the road surface without the blurring typically seen in previous methods, and the proposed method successfully generates images from novel views, outperforming earlier approaches. These findings demonstrate that the integrated depth information considerably enhances the quality of the novel view synthesis task.

4.3. Results on the JBNU-Depth360 Dataset

The method was also evaluated on a custom dataset collected from parking lot driving scenarios. The quantitative results of the view synthesis are presented in Table 2. Our proposed method has demonstrated significant improvements in both PSNR and SSIM metrics, achieving averages of 20.511 and 0.764, respectively. Furthermore, there was a notable reduction in LPIPS, which decreased from 0.541 of DS-NeRF to 0.529. This suggests that images rendered by the proposed method are more realistic compared to those synthesized by previous depth supervision approaches. We demonstrated the method in both public datasets and real-world data, including both indoor and outdoor environments.

Table 2. Quantitative results on the JBNU-Depth360 dataset. Higher is better for PSNR and SSIM and lower is better for LPIPS.

Method	PSNR↑					SSIM↑					LPIPS↓				
	Drive_2	Drive_3	Drive_4	Drive_5	Average	Drive_2	Drive_3	Drive_4	Drive_5	Average	Drive_2	Drive_3	Drive_4	Drive_5	Average
Nerfacto [30]	20.102	19.811	20.155	19.002	20.388	0.763	0.737	0.754	0.749	0.763	0.561	0.544	0.524	0.565	0.542
UrbanNeRF [9]	20.130	19.822	20.164	19.170	20.452	**0.764**	0.737	0.754	0.750	0.763	0.563	0.539	0.521	0.561	0.540
DS-NeRF [21]	20.157	19.819	20.341	19.071	20.468	0.763	0.738	0.754	0.750	0.763	0.564	0.540	0.521	0.568	0.541
Ours	**20.177**	**20.028**	**20.356**	**19.214**	**20.511**	0.763	**0.739**	**0.755**	**0.755**	**0.764**	**0.553**	**0.526**	**0.509**	**0.556**	**0.529**

Figure 4 presents a qualitative comparison of novel view synthesis using the JBNU-Depth360 dataset. In the first row, it is shown that previous methods often produce blurry results, particularly in floor details. In contrast, the proposed method effectively captures these details, reproducing them with higher fidelity to the ground truth data. The results in the second row further demonstrate that the proposed approach preserves even intricate structures. These findings suggest that integrating accurate sensor depth with densely inferred depth information leads to a more realistic synthesis.

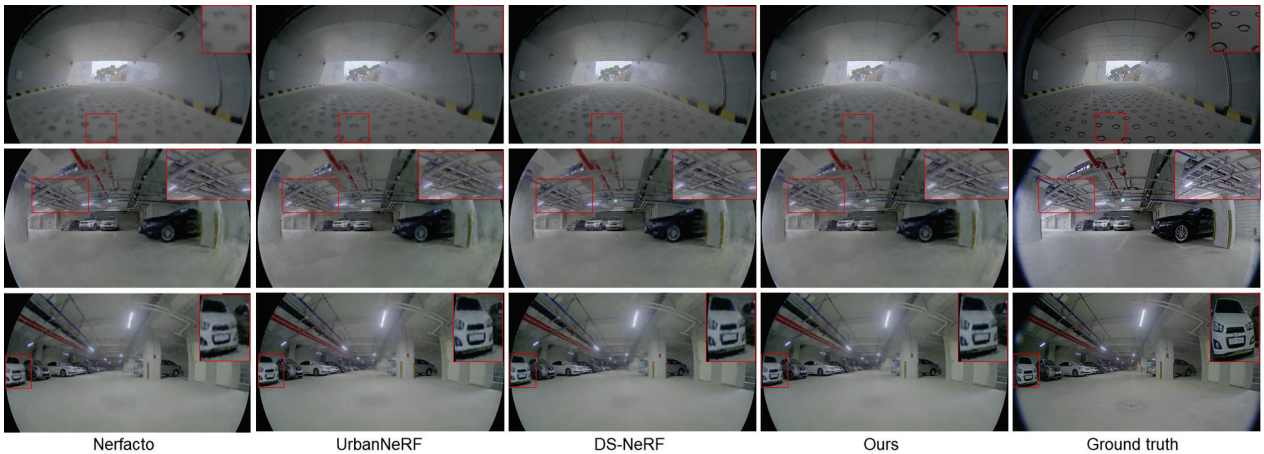


Figure 4. View synthesis on JBNU-Depth360. We have highlighted the details of the synthesized image with a red box.

4.4. Ablation Study

- Depth type comparison.

We investigate the impact of different depth types in Table 3. Both LiDAR projections and estimated depth maps provide advantages when optimizing the NeRF. Despite having fewer valid pixels, sparse LiDAR projections show similar PSNR and higher SSIM than the prior estimated depth. This result indicates that the accurate sensor data guided the NeRF model in learning the ray termination distribution and enhancing scene rendering. However, the dense depth map from the depth estimation model outperforms the view synthesis results from LiDAR data in terms of LPIPS. This implies that it provides more comprehensive perceptual information to the NeRF, even with some inherent uncertainties. Consequently, our approach demonstrated superior performance across all metrics, suggesting that the method of depth integration contributed significantly to optimizing the NeRF for realistic scene rendering.

Table 3. Effect of depth type including LiDAR and estimated depth. We compare different types of depth information using the KITTI-360 and JBNU-Depth360 datasets.

	Depth Type	**PSNR↑**	**SSIM↑**	**LPIPS↓**
KITTI-360	LiDAR projection	13.240	0.538	0.584
	Estimated depth	13.555	0.539	0.578
	Integrated depth	**13.678**	**0.548**	**0.557**
JBNU-Depth360	LiDAR projection	20.468	0.763	0.541
	Estimated depth	20.401	0.761	0.542
	Integrated depth	**20.511**	**0.764**	**0.529**

- Depth integration method comparison.

The edge-aware integration method is proposed to optimally leverage the sparse LiDAR data and prior knowledge of the monocular depth estimation model. In Table 4, we present the results of an ablation study that compares the spatial Gaussian function and the proposed integration method. Across all scenarios, the proposed approach consistently outperformed previous methods, achieving LPIPS of 0.574 and 0.529 for the KITTI-360 and JBNU-Depth360 datasets, respectively. These results indicate that considering the actual depth range is important when combining depth information to avoid depth differences, such as background and vehicles in the scene. Figure 5 shows the view synthesis results obtained from both datasets. We demonstrate that our approach helps preserve edge details by being aware of both spatial and depth range information, which helps NeRF generate realistic images.

Table 4. Integration method comparison. We conducted an ablation study on the integration method in terms of novel view synthesis using the KITTI-360 and JBNU-Depth360 datasets.

	Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
KITTI-360	Spatial Gaussian	13.502	0.533	0.592
	Proposed method	**13.678**	**0.548**	**0.557**
JBNU-Depth360	Spatial Gaussian	19.899	0.753	0.549
	Proposed method	**20.511**	**0.764**	**0.529**

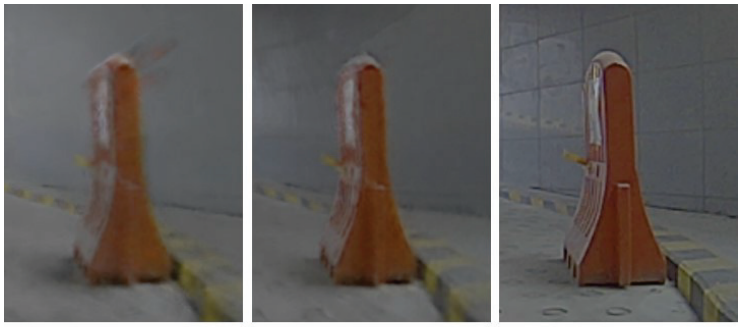


Figure 5. Qualitative results of an ablation study on edge-aware integrated function. Compared to the spatial Gaussian function, the proposed approach better preserves the object's edge information, resulting in a more realistic representation in the synthetic image.

5. Conclusions

In this work, we introduce a novel approach for synthesizing driving scenarios by leveraging the depth information. The NeRF's performance significantly degrades when dealing with fisheye-driving images due to the challenges of optimizing for their unbounded and distorted characteristics. It is important to note that driving data presents additional difficulties as it deviates from conventional object-centric datasets. To address these challenges, we propose the edge-aware integration loss that minimizes the discrepancy between the ray termination and integration of LiDAR data with densely estimated depth distribution. We evaluated the proposed method on the KITTI-360 and real-world JBNU-Depth360 datasets, achieving improvements of 0.438 in PSNR and 0.027 in LPIPs, respectively. It demonstrates that the method outperforms previous approaches in rendering novel views. However, the model performs poorly in cases of large rotations of the vehicle, where the number of scenes captured by the camera is relatively small, indicating the need for improvements in few-shot scenarios. The proposed model was validated on a limited dataset due to the scarcity of publicly available fisheye driving datasets. Future research will require the collection of fisheye camera datasets in more diverse driving scenarios. We hope that our work will inspire further research in fisheye rendering, and contribute to the advancement of autonomous driving tasks.

Author Contributions: Conceptualization, J.C.; methodology, J.C.; software, J.C.; validation, J.C.; data curation, J.C.; writing—original draft preparation, J.C.; writing—review and editing, J.C. and S.J.L.; visualization, J.C.; supervision, S.J.L.; project administration, S.J.L.; funding acquisition, S.J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT)(IITP-2024-RS-2024-00439292. This work was supported by the Materials/Parts Technology Development Program (20023305, Development of intelligent delivery robot with Cloud-Edge AI for last mile delivery between nearby multi-story buildings) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The KITTI-360 dataset is publicly available online, and it can be found at https://www.cvlibs.net/datasets/KITTI-360, accessed on 17 October 2024. The JBNU-Depth360 dataset is available at https://github.com/EunjinSon1/JBNU-Depth360, accessed on 17 October 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
2. Martin-Brualla, R.; Radwan, N.; Sajjadi, M.S.; Barron, J.T.; Dosovitskiy, A.; Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7210–7219.
3. Liu, S.; Zhang, X.; Zhang, Z.; Zhang, R.; Zhu, J.Y.; Russell, B. Editing conditional radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5773–5783.
4. Bao, C.; Zhang, Y.; Yang, B.; Fan, T.; Yang, Z.; Bao, H.; Zhang, G.; Cui, Z. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 20919–20929.
5. Lin, C.H.; Ma, W.C.; Torralba, A.; Lucey, S. Barf: Bundle-adjusting neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5741–5751.
6. Jeong, Y.; Ahn, S.; Choy, C.; Anandkumar, A.; Cho, M.; Park, J. Self-calibrating neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5846–5854.
7. Zeng, Y.; Lei, J.; Feng, T.; Qin, X.; Li, B.; Wang, Y.; Wang, D.; Song, J. Neural Radiance Fields-Based 3D Reconstruction of Power Transmission Lines Using Progressive Motion Sequence Images. *Sensors* **2023**, *23*, 9537. [CrossRef] [PubMed]
8. Ge, H.; Wang, B.; Zhu, Z.; Zhu, J.; Zhou, N. Hash Encoding and Brightness Correction in 3D Industrial and Environmental Reconstruction of Tidal Flat Neural Radiation. *Sensors* **2024**, *24*, 1451. [CrossRef]
9. Rematas, K.; Liu, A.; Srinivasan, P.P.; Barron, J.T.; Tagliasacchi, A.; Funkhouser, T.; Ferrari, V. Urban radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12932–12942.
10. Xie, Z.; Zhang, J.; Li, W.; Zhang, F.; Zhang, L. S-nerf: Neural radiance fields for street views. *arXiv* **2023**, arXiv:2303.00749.
11. Deng, L.; Yang, M.; Qian, Y.; Wang, C.; Wang, B. CNN based semantic segmentation for urban traffic scenes using fisheye camera. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 231–236.
12. Son, E.; Choi, J.; Song, J.; Jin, Y.; Lee, S.J. Monocular Depth Estimation from a Fisheye Camera Based on Knowledge Distillation. *Sensors* **2023**, *23*, 9866. [CrossRef] [PubMed]
13. Liao, Y.; Xie, J.; Geiger, A. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3292–3310. [CrossRef] [PubMed]
14. Barron, J.T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P.P. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5855–5864.
15. Tang, X.; Yang, M.; Sun, P.; Li, H.; Dai, Y.; Zhu, F.; Lee, H. PaReNeRF: Toward Fast Large-scale Dynamic NeRF with Patch-based Reference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 5428–5438.
16. Li, H.; Zhang, D.; Dai, Y.; Liu, N.; Cheng, L.; Li, J.; Wang, J.; Han, J. GP-NeRF: Generalized Perception NeRF for Context-Aware 3D Scene Understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 21708–21718.
17. Gu, K.; Maugey, T.; Knorr, S.; Guillemot, C. Omni-nerf: Neural radiance field from 360 image captures. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
18. Kulkarni, S.; Yin, P.; Scherer, S. 360fusionnerf: Panoramic neural radiance fields with joint guidance. *arXiv* **2022**, arXiv:2209.14265.
19. Choi, J.; Hwang, G.; Lee, S.J. DiCo-NeRF: Difference of Cosine Similarity for Neural Rendering of Fisheye Driving Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 7850–7858.
20. Wei, Y.; Liu, S.; Rao, Y.; Zhao, W.; Lu, J.; Zhou, J. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5610–5619.
21. Deng, K.; Liu, A.; Zhu, J.Y.; Ramanan, D. Depth-Supervised NeRF: Fewer Views and Faster Training for Free. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12882–12891.
22. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
23. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.

24. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
25. Wang, G.; Chen, Z.; Loy, C.C.; Liu, Z. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 9065–9076.
26. Wang, C.; Sun, J.; Liu, L.; Wu, C.; Shen, Z.; Wu, D.; Dai, Y.; Zhang, L. Digging into depth priors for outdoor neural radiance fields. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 1221–1230.
27. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Proceedings, Part V 12; Springer: Berlin/Heidelberg, Germany, 2012; pp. 746–760.
28. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
29. Lee, J.H.; Han, M.K.; Ko, D.W.; Suh, I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv* **2019**, arXiv:1907.10326.
30. Tancik, M.; Weber, E.; Ng, E.; Li, R.; Yi, B.; Kerr, J.; Wang, T.; Kristoffersen, A.; Austin, J.; Salahi, K.; et al. Nerfstudio: A modular framework for neural radiance field development. *arXiv* **2023**, arXiv:2302.04264.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Review

A Comprehensive Evaluation of Iris Segmentation on Benchmarking Datasets

Mst Rumana Sumi [1,*,†], Priyanka Das [1,†], Afzal Hossain [1], Soumyabrata Dey [2] and Stephanie Schuckers [1]

[1] Department of ECE, Clarkson University, Potsdam, NY 13699, USA; afhossa@clarkson.edu (A.H.); sschucke@clarkson.edu (S.S.)
[2] Department of Computer Science, Clarkson University, Potsdam, NY 13699, USA
* Correspondence: sumima@clarkson.edu
† These authors contributed equally to this work.

Abstract: Iris is one of the most widely used biometric modalities because of its uniqueness, high matching performance, and inherently secure nature. Iris segmentation is an essential preliminary step for iris-based biometric authentication. The authentication accuracy is directly connected with the iris segmentation accuracy. In the last few years, deep-learning-based iris segmentation methodologies have increasingly been adopted because of their ability to handle challenging segmentation tasks and their advantages over traditional segmentation techniques. However, the biggest challenge to the biometric community is the scarcity of open-source resources for adoption for application and reproducibility. This review provides a comprehensive examination of available open-source iris segmentation resources, including datasets, algorithms, and tools. In the process, we designed three U-Net and U-Net++ architecture-influenced segmentation algorithms as standard benchmarks, trained them on a large composite dataset (>45K samples), and created 1K manually segmented ground truth masks. Overall, eleven state-of-the-art algorithms were benchmarked against five datasets encompassing multiple sensors, environmental conditions, demography, and illumination. This assessment highlights the strengths, limitations, and practical implications of each method and identifies gaps that future studies should address to improve segmentation accuracy and robustness. To foster future research, all resources developed during this work would be made publicly available.

Keywords: biometric; iris; segmentation; deep learning

1. Introduction

A traditional iris recognition system includes four basic steps: [1]—(1) iris image acquisition, (2) segmentation, (3) normalization, and (4) feature extraction and matching. Iris segmentation involves locating and isolating the iris from other regions of the eye, including noise detection like occlusion from eyelashes, eyelids, or hair. Thus, it plays an important role in the recognition system. When the iris segmentation fails, even the best feature extraction method will be incapable of generating an iris code that corresponds to the actual iris texture, increasing the chances of false rejection [2]. Iris image quality and acquisition conditions generally affect the accuracy of this segmentation task. An iris image captured in less-constrained environments may include noises, such as motion blur, eyelids or eyelash occlusion, eyeglass occlusions, off-angle iris, specular reflections, etc., that make iris segmentation more challenging.

Conventionally, iris images are captured under NIR illumination for biometric recognition. However, there is an interest in the community to develop iris recognition technologies under visible illumination (VIS) [3,4]. Traditional NIR-based iris segmentation methods are mostly based on Daugman's integro-differential operator [5], Wildes' canny operator, or Hough transform [6]. These methods are suitable for cooperative or highly controlled iris biometric setups such as close acquisition distance, stop-and-stare verification, image capture

under near-infrared illumination, etc. Traditional methods are not suitable for noisy data and data captured under a non-cooperative environment or less controlled environments such as acquisition at a distance, on the move, and image capture under visible illumination [7].

With the rapid increase of biometric recognition applications, there is a demand to update iris recognition technology, making it more noise-tolerant and user-friendly. Recently, deep learning (DL)-based iris segmentation methods have been introduced in research studies to overcome the shortcomings of traditional approaches [1,2,7,8]. However, open-source DL-based iris segmentation models are sparse and have limited reproducibility, limiting usage by the community and continuity for further improvement. Furthermore, one crucial requirement for developing efficient DL models is the availability of accurate ground truth iris masks in large volumes, and the unavailability of that is a significant limitation. Hofbauer et al. [9] made strides by publicly releasing two ground truth datasets, IRISSEG-CC [10] and IRISSEG-EP [11]. Nonetheless, a notable drawback of these datasets is their omission of eyelash annotations, which compromises the ability of models trained on them to detect eyelash occlusions, adversely affecting the performance of iris recognition systems.

In the literature, several iris segmentation techniques have been proposed, with reported success and limitations. However, some of the challenges, which are yet to be addressed, are listed below:

- Datasets used for the development of algorithms are often proprietary and lack crucial labels for features such as eyelashes, which are essential for accurate segmentation.
- The scarcity of open-source codes hinders the reproducibility of these methods.
- Algorithms are often evaluated on the same dataset they were trained on, limiting cross-dataset performance analysis, which is crucial for generalization and robustness in real-world applications.

This study attempts to identify the gaps in resources for iris segmentation and benchmark the open-source algorithms (both traditional and DL-based) against common datasets for direct comparison. In the process, the performance of the algorithms is evaluated based on several criteria:

- **Sensitivity:** The algorithms are assessed for their sensitivity toward different sensors, environments, and demographic variations. This involves a multi-dataset evaluation to understand how well the algorithms can adapt to various conditions.
- **Generalization capability:** The cross-dataset performance of the algorithms is examined to determine how well an algorithm can generalize its learning from one dataset to another, which may have different characteristics.
- **Practicability:** The practical aspects of the algorithms are also evaluated, including the inference time, the number of parameters, and the size of the model. These factors contribute to the feasibility of deploying the algorithm in real-world applications.
- **Noise detection capability:** The ability of the algorithms to detect and handle noise, such as eyelash detection, is assessed. This capability is important for improving the accuracy of the algorithms.
- **Illumination:** The performance of the algorithms under different lighting conditions, such as Near-Infrared (NIR) and Visible Spectrum (VIS), is evaluated. This helps in understanding the versatility and robustness of the algorithms in varying illumination scenarios.

In addition to the thorough comparative analysis of the open-source iris segmentation algorithms, we developed an end-to-end iris segmentation resource set with the motivation to append to the fragmented existing public resources. The resource set consists of the following: 1. A *composite iris dataset of 45k+ samples*, gleaning publicly available datasets along with corresponding iris masks generated using OSIRISv4.1 (Open Source Iris Recognition Software) [12]; 2. A GUI-based *iris segmentation toolkit* to generate iris masks (ground truth); 3. A set of *1K challenging ground truths*, including labels for eyelashes, hand-crafted using the GUI; and 4. *Three iris segmentation models* influenced by U-Net architecture trained on the composite dataset and fine-tuned on the handcrafted ground truth. The main contributions of our study are as follows:

- **Benchmarking of eleven iris segmentation methods** against **five datasets** of different compositions (illumination, sensor, noise level, demography), providing consistent and fair comparisons that offer valuable insights into real-world performance.
- **Development of a manual segmentation toolkit** and preparation of **1k manually segmented iris masks** of challenging samples such as iris images with eyelashes, eyeglasses, eyelid occlusion, and low usable iris area.
- **Development of three iris segmentation benchmarked models** influenced by state-of-art U-Net [13] and U-Net++ [14] architectures (average accuracy $>$ 90%), have generalized learning ability (cross-dataset accuracy $>$ 90%), are efficient in detecting eyelashes (average accuracy $>$ 90%), and perform equally well on NIR and VIS datasets.
- **Open-source** the resource set for the benefit of the research community, with the exception of the composite dataset for licensing restrictions. However, the composite dataset is reproducible—the datasets are publicly available, and OSIRIS is open-source. The resource set is available at https://github.com/RumanaSum/Iris-Segmentation (accessed on 24 February 2024).
- A comprehensive evaluation that highlights the strengths and limitations of the models, establishing a standard foundation for future research in iris segmentation and facilitating the development of highly accurate and robust benchmark models to serve as reliable references for further advancements in the field.

The rest of the paper is organized as follows: Section 2 introduces the related work on iris segmentation. Section 3 provides dataset preparations and implementation details of the developed models. Section 4 presents the results and analyzes the performance of the implemented methods. Section 5 discusses the limitations and road map for iris segmentation. The glossary section provides the glossary of terms used in this paper.

2. State of the Art

Multiple DL-based iris segmentation models have been proposed in the recent past. Following an assessment of the relevant work, the limitations in the scope of iris segmentation, such as availability of the models for reusability or reproducibility, availability of ground truth data for re-training, and eyelash detection capability, are summarized in Table 1.

Table 1. The state-of-the-art iris segmentation algorithms.

Research	Resource Availability			
	Code	**Training Mask**	**Test Ground Truth**	**Eyelash Marking**
Miron et al. [15]				
Hou et al. [8]				
Yang et al. [16]				
Meng et al. [17]				
Lozej et al. [18]	✓			✓
Zang et al. [19]		✓		
Jalilian et al. [20]		✓		
Li et al. [21]		✓	✓	
Chen et al. [22]		✓		
Wang et al. [7]	✓	✓	✓	✓
Wang et al. [16]				
Wang et al. [23]		✓	✓	✓
Bezerra [1]		✓	✓	✓
Trokielewicz et al. [24]	✓	✓		
Proposed work	✓	✓	✓	✓

Bezerra et al. [1] proposed GAN and a FCN-based iris segmentation method and evaluated them on NIR and VIS datasets. The source code or network weights of these models are not open-source. The authors publicly shared manually annotated 1K images from CASIA Thousand [25], 1K from CrEye-Iris [26], and 431 from MICHE-I [27] datasets. Lozej et al. [18] trained a U-Net [13]-based model on 200 manually labeled ground truths from the CASIA Interval dataset [25]. While the model weights are publicly available, the ground truth remains proprietary. Wang et al. [7] proposed a multi-tasking U-Net segmentation model that can generate iris masks, iris outer boundary, and pupil masks from the raw iris image. For training this model, the authors employed publicly available iris masks along with self-annotated pupil masks and iris outer boundary masks. The model is evaluated on CASIA Distance [25], UBIRIS.v2NICE.I [28], and MICHE-I [27] datasets. Their self-annotated ground truth masks and network weights are publicly available. Trokielewicz et al. [24] fine-tuned a SegNet architecture with 1300 self-annotated postmortem iris images obtained from the Warsaw-BioBase-Post-Mortem-Iris v1.0 [29] dataset. Their annotated ground truths and source code were also released for further experimentation. Chen et al. [22] independently trained a CNN-based model on the CASIA Interval [25] and IIT Delhi Iris [30] datasets with self-annotated and publicly available ground truths (obtained from the IRISSEG-EP dataset). The self-annotated ground truths and the models are not public. Hofbauer et al. [9] publicly shared two ground truth datasets—IRISSEG-CC [10] and IRISSEG-EP [11]. One limitation of these datasets is that the eyelashes are not marked. Thus, the models trained on these datasets would lack the capability to detect eyelash occlusions, resulting in a negative impact on the performance of the subsequent iris recognition system. The authors [22] cautioned of misleading higher segmentation accuracy using the IRISSEG-EP [9] dataset as it excludes eyelash labels. A few other proposed iris segmentation studies provide no information regarding the availability of self-annotated ground truths, models, or network weight [8,15–17,19,20]. Huo et al. [8] proposed an Attention Mechanism U-Net++, employing a pruning scheme to obtain four iris segmentation networks. Wang et al. [16] developed a lightweight, FCN-based network. They adopted multi-level feature-dense fusion modules, multi-supervised training of multiscale images, and generative adversarial networks to improve the segmentation performance. Meng et al. [17] combined Swin-T with CNNs and introduced a bilateral segmentation backbone network. Zhang et al. [19] combined the dilated convolution with the U-Net to extract more global features. Jalilian et al. [20] proposed three types of FCN-based networks. Miron et al. [15] proposed a modified lightweight U-Net-based architecture. Lian et al. [31] added an attention module to the U-net to increase the weight of the iris. Despite the importance of eyelash labels on the training mask, only a few models in the literature [1,7,18,23] were trained with eyelash labels. The models proposed in [2,19,20] were trained on a public ground truth dataset [9], which does not have annotated eyelashes. The source of ground truth for training other models [8,15–17] was not mentioned; eyelash annotations were also not found by our visual inspection. The availability of more annotated iris masks with eyelash labels will strengthen the iris segmentation research. The review of related works also indicates that a thorough comparative analysis of all available iris segmentation methods is required on reference datasets to pinpoint their effectiveness and limitations. This analysis promotes transparency, reproducibility, and innovation, enabling the identification of gaps in current methodologies and guiding future advancements in the field.

Addressing these gaps, we developed three new open-source models based on U-Net and U-Net++ as benchmarks and created a manual segmentation toolkit along with a manually annotated ground truth dataset that includes eyelash annotations. Additionally, our study performed a comprehensive analysis of eight existing iris segmentation methods—comprising two traditional approaches (OSIRIS [12], USIT (University of Salzburg Iris Toolkit) [32]) and six DL-based approaches ([1,7,18,23,24] selected based on the availability of their source code and the inclusion of eyelash annotations in their ground truths (Table 1). This analysis was conducted using our benchmark models across multiple

datasets to evaluate their effectiveness and limitations. Our approach not only addresses current shortcomings in the literature but also establishes a new standard for advancing iris segmentation studies.

3. Experiments

Eleven state-of-the-art iris segmentation methods are studied on five datasets for evaluating generalizability, practicability, noise detection capability, and sensitivity towards illumination. In the process, we developed three benchmark iris segmentation models based on U-Net and U-Net++ architecture, prepared a large composite NIR iris dataset by combining multiple public datasets, created a manual iris segmentation toolkit, developed a manually annotated iris-mask ground truth dataset with eyelash annotations for fine-tuning, and adapted eight other algorithms (two traditional and six DL-based) from recent literature, chosen for their accessible source code and ground truth that includes eyelash labels (Table 1). For the purpose of performance evaluation, we focused on studies where the source code or detailed ground truth for iris masks with eyelash annotations was available. Studies lacking in these respects were excluded from our re-evaluation process. The implementation details of all the methods are discussed in this section.

3.1. Iris Segmentation Toolkit Development

In systems reliant on machine learning, the quality of the input directly influences the quality of the output. Specifically, in the context of iris recognition, the accuracy of iris segmentation and the ability to detect noise are pivotal for the success of the entire recognition process. Thus, for a DL-based iris segmentation model, highly precise ground truth is vital for training. This presents two significant challenges: the scarcity of publicly available, large-scale, high-quality ground truth data, and among the datasets that are available, a limited number include annotations for eyelashes. While OSIRIS, a traditional open-source segmentation tool that employs an integro-differential operator, serves as a resource for generating ground truth, the accuracy of its generated masks is often compromised by challenges such as segmentation errors in cases of low usable iris area, constricted pupils, high illumination, and inaccuracies in eyelash and pupil boundary detection. Our thorough review of existing literature revealed a lack of tools, either manual or automated, dedicated to the generation of iris segmentation masks. Recognizing the critical need for such a resource, we developed a segmentation and annotation toolkit specifically designed for iris segmentation tasks. This toolkit utilizes a pair of ellipses to delineate the iris's inner and outer boundaries and allows for pixel-level manual annotation to accurately identify eyelids, eyelashes, and other non-iris eye components. The user interface of the toolkit is illustrated in Figure 1. Figure 2 shows an example of annotated ground truth. The toolkit, its features, and the guidance for usage are available in our GitHub repository.

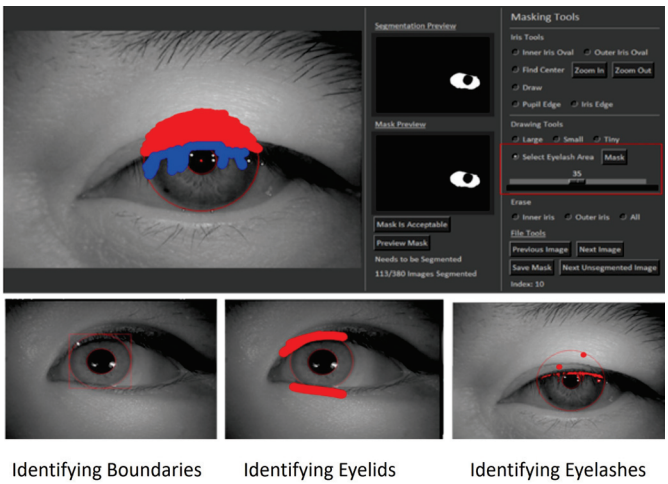


Figure 1. User interface of the iris segmentation toolkit.

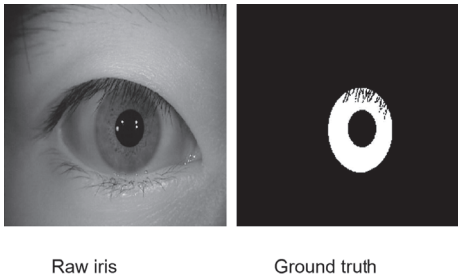


Figure 2. Example of annotated ground truth using the toolkit.

3.2. Dataset Preparation

We employ the following five datasets for our experiments. The statistics for each dataset is presented in Table 2.

Table 2. Summary of all datasets used in this study.

Dataset	Total Images	Train Samples	Test Samples	Resolution	Wavelength
Composite	45,683	41,283	4400	640 × 480	NIR
Composite Subset	1006	616	330	640 × 480	NIR
CASIA Distance	400	300	100	640 × 480	NIR
CASIA Thousand	1000	800	200	640× 480	NIR
UBIRIS.v2NICE.I	915	479	428	400 × 300	VIS
MICHE-I	871	680	191	Various	VIS

(A) Composite dataset: To address the lack of large annotated iris dataset for the training, we prepared a composite NIR dataset by combining multiple publicly available NIR datasets—CASIA Iris Lamp [25], CASIA Iris Twins [25], CASIA Iris Thousand [25], and ITR Iris Clarkson [33] datasets. Because of the large volume of this dataset, manual segmentation of all the samples would be a daunting task. Instead, Daugman-style open-source OSIRIS [12] was used to generate masks. Segmentation errors by OSIRIS were manually identified and removed. Additionally, VeriEye [34] matcher was used to determine segmentation errors based on the match score (a subset of false accepts and rejects); the errors were removed. The final composite dataset containing 45,683 samples from 1852 subjects is summarized in Table 3. Additionally, we manually annotated 1K samples (referred to as the composite subset dataset for the rest of the paper) using our developed toolkit from the erroneous segmented samples. For visual analysis, we also prepared a separate test set of 60 challenging samples, which OSIRIS failed to segment accurately, leading to a false rejection.

Table 3. An overview of the composite dataset.

Dataset	No of Subjects	Unique ID	No of Samples
CASIA Iris Thousand	1000	2000	19,338
CASIA Iris Twin	200	400	3082
CASIA Iris Lamp	410	820	16,081
ITR Clarkson	242	479	7182
Total	1852	3699	45,683

(B) UBIRIS.v2 NICE.I [28]: A subset of the UBIRIS.v2 dataset containing manually labeled iris masks used in NICE.I competition [35] was made available by [7]. The subset contains approximately 1K samples, captured on the move and at a distance with a Canon

EOS 5D camera under visible illumination at a resolution of 400 × 300 pixels. We utilized the same training and testing set as [7].

(C) MICHE-I [27]: MICHE-I was created from 92 subjects by three mobile devices—the iPhone 5, Samsung Galaxy S4, and Samsung Galaxy Tab2—in uncontrolled conditions with visible illumination. A subset of 871 samples from this dataset was compiled by [7] from the original manual annotations by [1,36]. We utilized the training and testing set of [7].

(D) CASIA Iris Thousand [25]: We used this entire dataset with the OSIRIS-generated masks to prepare the composite dataset. For our experiment, we also used a subset of 1K samples with manual ground truths shared by [1]. We separately trained all models on these 1K ground truths. Following [1], we utilized 80% for training and 20% for testing.

(E) CASIA Distance [25]: CASIA Distance contains 2567 images from 142 subjects captured from a three-meter distance with a CASIA long-range iris camera under NIR illumination. A subset of 400 manually annotated iris masks of resolution 640 × 480 pixels from the first 40 subjects was publicly shared by Liu [37]. Like Wang et al. [7], we used the subset from [37], consisting of 400 iris images with manually labeled iris masks. Following [7], the first 300 images were used for our training, and the last 100 were used for testing.

3.3. Iris Segmenter Design

U-Net Architecture: Our proposed benchmark model, illustrated in Figure 3, is based on a U-Net [13] architecture. The encoder path of this architecture consists of a repeated implementation of two 3 × 3 convolutional layers and a ReLU activation layer with a 2 × 2 max pooling operation with stride 2 for down-sampling. Thus, each down-sampling step doubles the number of feature channels and decreases the image resolution by half. Each step in the decoder path of the architecture consists of a 2 × 2 up-convolution, a concatenation with cropped feature map from the encoder path, and two 3 × 3 convolutions, followed by a ReLU activation layer. Thus, each up-sampling step (opposite to down-sampling) doubles the image resolution and decreases the number of feature channels by half. The concatenation ensures the propagation of all information from the encoder to the decoder and no loss of information while down-sampling. The final layer of the network is a 1 × 1 convolutional layer that combines the preceding layer's output and produces the segmentation maps.

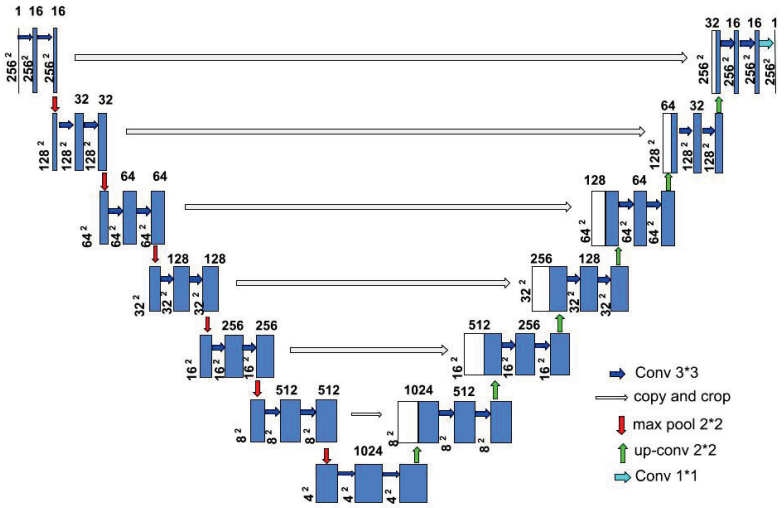


Figure 3. Architecture of the proposed U-Net benchmark model.

U-Net++ Architecture: Figure 4 shows our second benchmark model based on the U-Net++ architecture [14]. Similar to U-Net architecture, the encoder consists of two repeated 3 × 3 convolution layers, each followed by a ReLU activation layer and a 2 × 2 max-pooling for down-sampling. Unlike the U-Net model, the encoder and decoder are

connected through a series of nested dense convolutional blocks, which bridge the semantic gap between the feature maps of the encoder and decoder prior to fusion. Each block in the decoder combines the multiscale feature maps passed horizontally from its preceding blocks (with the same image resolutions) as well as the multiscale feature maps passed vertically (different image resolutions); details are in [14]. Each step in the decoder path also consists of a 2×2 up-convolution, concatenation of the preceding multiscale feature map, and two 3×3 convolutions, each followed by a ReLU activation layer. Finally, the output of the last decoder passes through a 1×1 convolution with sigmoid activation and produces the segmentation mask.

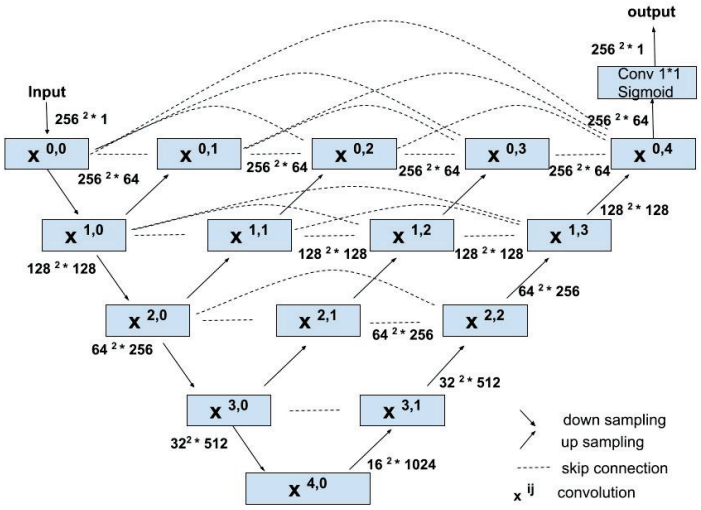


Figure 4. Architecture of the proposed U-Net++ benchmark model.

3.4. Implementation Details

U-Net and U-Net++ (ours): Our proposed benchmark models, illustrated in Figures 3 and 4, are based on U-Net [13] and U-Net++ architecture [14], respectively. We implemented the models in Python using Keras API with TensorFlow as its backend. The training dataset was augmented with rotation, zoom, width, and height shifts to improve the model's generalization capability. The models are trained with the image resolution of 256×256, batch size of 8, Adam optimizer with a learning rate of 10^{-4}, binary crossentropy loss, and early stopping.

The training procedure was divided into two distinct phases. Initially, the base models, either U-Net or U-Net++, were trained on our composite dataset, which included 45.6k samples (with 90% allocated for training and 10% for validation) using masks generated by the OSIRIS software. This initial phase aimed to establish a robust foundational model. In the subsequent phase, these pre-trained models were fine-tuned on various other datasets to evaluate their adaptability and performance across different imaging conditions and challenges. This included fine-tuning on datasets such as the composite subset (with 1K manually annotated masks), CASIA.v4-Distance [25], CASIA Iris Thousand [25], UBIRIS.v2 NICE.I [28], and MICHE-I [27] datasets, each selected for their unique attributes related to demography, NIR and VIS imaging, sensor types, collection methodologies, and environmental contexts.

The rationale behind this two-step training approach was to first leverage the large volume of data in the composite dataset for foundational training, despite the potential limitations of OSIRIS-generated masks in accurately capturing challenging aspects like eyelashes, eyelids, and eyeglasses, or areas of smaller usable size. The initial training phase potentially limited the models' performance on such challenging samples. Therefore, fine-tuning the models on manually annotated masks was intended to specifically improve their ability to accurately identify and handle these intricate scenarios, thereby enhancing their overall performance and applicability across a broader range of real-world conditions.

U-Net with weight map (ours): In addition to the implementation procedure followed in the U-Net and U-Net++, the U-Net base model was finetuned with the weight-mapped ground truth to increase the weight of the eyelash pixels and draw increased attention to the network to learn eyelash detection, following [13] and weighted binary cross-entropy loss, computed as *weight map* × *binary crossentropy loss*.

Lozej et al. [18]: The authors developed a U-Net-based segmentation method, experimented with different hyperparameter settings, and trained the model on 200 samples (160 samples for training and 40 for testing) from CASIA Iris Interval [25]. The best model weight was made publicly available without the source code. We re-implemented the method following [18]. We used their original pre-trained model as the base model and fine-tuned all the layers independently on five datasets (mentioned above). The authors trained their model for ten epochs. Similarly, Like [18], we trained the model for ten epochs. Additionally, we conducted a separate training using early stopping for a fair comparison.

Trokielewicz et al. [24]: The authors developed a postmortem segmentation model based on the SegNet architecture. SegNet consists of an encoder (a pre-trained VGG-16 network excluding the classification layer) and a corresponding decoder, followed by a classification layer. The author initialized the training with pre-trained ImageNet weight [38] and retrained on 1300 manually annotated postmortem iris images obtained from the Warsaw-BioBase-Post-Mortem-Iris v1.0 [29] dataset. The source code is publicly available. We reproduced this method as the authors suggested and fine-tuned it independently on five different datasets (mentioned earlier), initializing with ImageNet weight.

Bezerra et al. [1]: The authors proposed two iris segmentation methods based on GAN and FCN. Due to the unavailability of source code, limiting reproducibility, we only report the results presented in the paper. However, the authors made their manually annotated ground truths publicly available. We used their manual ground truths for CASIA Iris Thousand [25] and Miche-I [27] datasets to fine-tune other methods implemented in this study.

Wang et al. [7]: The authors proposed a multitask U-Net iris segmentation model, which required an iris mask, pupil mask, and iris outer boundary as the input. The source code is publicly available, including manually annotated pupil masks and outer boundary masks for CASIA Distance [25], UBIRIS.v2 NICE.I [28], and MICHE-I [27] datasets. We did not implement their method in this study as the manual annotation of the pupil mask and iris outer boundary mask for our other test datasets would require additional resources. We reported their performance. We employed their datasets for other implemented methods.

Wang et al. [23]: Refs. [7,23] used the same network with a minimum modification. They also used the same two datasets. Hence, we only reported their accuracy.

The OSIRISv4.1 [12]: This open-source Dougman-style software comprises four key modules: segmentation, normalization, feature extraction, and matching. We compared the segmentation module with other implementations.

USIT [32]: The toolkit includes several traditional iris preprocessing, feature extraction, and feature comparison techniques. We used the Wahet (Weighted Adaptive Hough and Ellipsopolar Transform [39]) segmentation method to compare it with other implemented methods.

4. Results

4.1. Evaluation Metrics

The following metrics are used to evaluate the segmentation methods.

mIoU: Mean Intersection Over Union (mIoU) is a standard evaluation metric for semantic segmentation. IoU indicates the proportion of intersection and union of the ground truth and predicted segmentation. The mIoU is calculated as the mean value of the IoU of all iris images. The mIoU is defined as follows:

$$mIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{True Positive}_i}{\text{True Positive}_i + \text{False Positive}_i + \text{False Negative}_i} \quad (1)$$

F1 score: F1 score is the harmonic mean of the precision and recall. Precision is the ratio of the number of correctly classified iris pixels to the number of all pixels that are classified as iris pixels, including those not identified correctly. The recall is the ratio of the number of correctly classified iris pixels to the number of all iris pixels that should have been identified as the iris. F1 is calculated as follows:

$$F1\,\text{score} = \frac{2 \times \text{True Positive}}{2 \times \text{True Positive} + \text{False Positive} + \text{False Negative}} \quad (2)$$

F1 and mIoU scores are bounded between 0 and 1. A higher score represents better segmentation accuracy. Both the F1 score and mIoU are widely recognized and standard metrics in the field of iris segmentation for evaluating model accuracy, providing a reliable assessment of segmentation performance in biometric systems.

4.2. Performance Evaluation

We conducted a comprehensive quantitative and qualitative analysis of 11 state-of-the-art segmentation models, including the three benchmark models we developed. This thorough analysis involved intra-dataset performance assessments across five diverse datasets, taking into account various factors such as demography, NIR and VIS illumination, sensor types, collection setups, and environmental conditions. Furthermore, we assessed the models' ability to detect noise (eyelash detection), their cross-dataset performance, and practical usability.

In addition to intra-dataset performance, we placed emphasis on cross-dataset evaluation to assess the generalization capability of the models. This is a crucial aspect for biometric systems that need to operate effectively on unseen data in real-world applications. Our benchmarking methodology also accounts for practicability by evaluating inference time and computational efficiency, both of which are essential for real-time deployment in biometric systems. These factors together ensure that the models not only excel in accuracy but are also robust and practical for deployment.

(A) Intra-dataset performance evaluation: a detailed summary of intra-dataset performance is provided in Table 4, with significant insights for each dataset detailed below:

Composite subset (NIR): Our benchmark model, U-Net, achieved the highest mIoU score of 91.70% and the second-highest F1 score of 94.04%. Our U-Net++ model secured the second-highest mIoU of 91.41% and the highest F1 score of 94.20%. Lozej et al. (FT-10) and Lozej et al. (FT-ES) posted mIoU scores of 91.13% and 91.16%, with F1 scores of 93.62% and 94.01%, respectively. Our U-Net with weight map model achieved 91.03% mIoU and 93.80% F1 score. Trokielewicz et al. (FT), OSIRIS, and USIT performed with mIoU scores of 85.09%, 85.98%, and 87.30%, and F1 scores of 88.14%, 87.17%, and 87.10%, respectively. This indicates that these models are less suited for this dataset compared with the other DL-based models.

CASIA Thousand (NIR): Our U-Net++ models achieved the highest mIoU score of 95.26% and the second-highest F1 score of 95.22%. Our U-Net model also demonstrated comparable results. The models developed by Lozej et al. achieved mIoU scores similar to those of U-Net and U-Net++, albeit with a lower F1 score of 94.61%. The FCN and GAN models by Bezerra showed robust performance, with F1 scores of 94.42% and 95.38% (highest among all models), respectively, suggesting they are highly effective for this dataset. The performance of Trokielewicz et al. (FT) and traditional methods such as OSIRIS v4.1 and USIT (Wahet) was comparatively degraded, with IoU scores of 82.98%, 88.51%, and 80.83%, and F1 scores of 89.52%, 87.78%, and 81.62%, respectively.

CASIA Distance (NIR): Our U-Net++ model achieved the highest IoU and F1 scores of 94.72% and 94.51%, respectively. The U-Net model achieved a mIoU of 94.48% and an F1 score of 93.58%. The models of Lozej et al. (FT-10) and Lozej et al. (FT-ES) secure mIoU scores of 92.80% and 93.56% and F1 scores of 92.20% and 93.04%, respectively. Models developed by Wang [7] and Wang [23] both achieved F1 scores of 94.25% and 94.30%, respectively. However, Wang's [23] model achieved a lower mIoU score (89.40%) compared

with our models and Lozej's models. The model from Trokielewicz et al. (FT) recorded an IoU of 79.38% and an F1 score of 85.28%, highlighting its lesser effectiveness compared with the other deep learning (DL) models. The traditional method, OSIRIS v4.1, achieved an IoU of 83.38% and an F1 score of 82.87%, demonstrating that traditional approaches are less effective for this dataset in comparison to DL-based models. USIT (Wahet) achieved a mIoU of 70.34% and an F1 score of 72.45%, indicating its unsuitability for this dataset and further underscoring the superiority of DL-based methods for this dataset.

UBIRIS.v2 NICE.I (VIS): Our U-Net and U-Net++ models, along with Lozej's models, achieved F1 scores of 90.89%, 90.87%, and 90.66%, respectively. In terms of the IoU scores, our U-Net++ and U-Net models recorded the highest (91.86%) and second highest (91.78%) scores, respectively, while Lozej's model achieved a mIoU of 91.04%. Wang et al. reported the highest F1 score of 91.78% for this dataset, indicating top performance among the evaluated models. Additionally, Bezerra's GAN demonstrated strong performance with an F1 score of 91.42%. The performance of Bezerra's FCN, with an F1 score of 88.20%, and the Trokielewicz et al. (FT) models, with an F1 score of 85.28%, was less effective in segmentation compared with other deep learning models for this dataset. Traditional methods, such as OSIRIS v4.1 and USIT (Wahet), were found to be unsuitable for this dataset, achieving mIoU and F1 scores ranging from 20% to 43%, further underscoring the inadequacy of traditional methods for this dataset.

MICHE-I (VIS): Our benchmark models U-Net and U-Net++ stood out with the highest and second-highest mIoU scores of 92.98% and 92.94%, respectively, complemented by F1 scores of 92.27% and 92.82%. Lozej's model achieved a mIoU of 92.36% and a F1 score of 91.83%. Trokielewicz's model was less effective, achieving an IoU score of 84.21% and an F1 score of 83.42%. Bezerra's FCN and GAN models showed F1 scores of 83.03% and 87.20%, indicating they were less efficient than the U-Net variants. Traditional methods, OSIRIS v4.1 and USIT (Wahet), showed comparatively degraded performance, with F1 scores of 32.48% and 26.03%, respectively, underscoring their unsuitability for this dataset.

In summary, the performance of all U-Net variant models, including those developed by us, Lozej, and Wang, was comparable and consistent across both NIR and VIS datasets, with minimal differences in accuracy observed. The FCN model demonstrated strong performance on the CASIA Thousand dataset but was less effective on the UBIRIS.v2 NICE.I (VIS) and MICHE-I (VIS) datasets, indicating its reduced efficacy for VIS datasets. Similarly, GAN-based models exhibited robust performance for the CASIA Thousand and UBIRIS.v2 NICE.I (VIS) datasets but were less effective on the MICHE-I (VIS) dataset compared with the U-Net variants. DL-based iris segmentation methods, with the exception of the model developed by Trokielewicz et al. [24], consistently outperformed traditional methods such as OSIRIS [12] and USIT [32] across all evaluated metrics for each dataset. Trokielewicz et al. [24] surpassed traditional methods in all metrics on VIS datasets and achieved higher F1 scores, albeit with a lower mIoU score than OSIRIS, for NIR datasets. Initially designed for postmortem data, the model by Trokielewicz et al. [24] notably outperformed OSIRIS on postmortem datasets. However, our observations lead to the conclusion that a model tailored for postmortem iris analysis can not be optimally fine-tuned for live iris datasets, resulting in its under-performance compared with more recent DL-based models and traditional iris segmentation methods on NIR datasets with live subjects. These findings further underscore that traditional methods are not well-suited for VIS datasets.

(B) Assessment of eyelash detection capability: To evaluate the models' capability in eyelash detection, we separated 160 samples with eyelash occlusions from the test set of the composite subset dataset (refer to mIoU EL in Table 4). mIoU was used as an evaluation metric. mIoU refers to the score for the entire test set of 330 samples. Our three models and Lozej's model demonstrated strong (>90%) accuracy and comparable performance in eyelash occlusion detection. Despite these high accuracy rates, a decline in performance was noted for the models when dealing with samples containing eyelash occlusions, as opposed to their performance on the complete general test set. Limited training iris masks

with annotated eyelashes could be the reason for the lower scores. Adding more training masks with annotated eyelashes may improve this performance.

Table 4. Performance evaluation of the state-of-the-art models on five benchmarking datasets.

Method	Composite Subset (NIR)					Casia Thousand (NIR)			Casia Distance (NIR)			UBIRISv2NICE.I (VIS)			MICHE-I (VIS)		
	Th	mIoU (0-1)	F1 (%)	mIoU EL	F1 (%) EL	Th	mIoU (0-1)	F1 (%)	Th	mIoU (0-1)	F1 (%)	Th	mIoU (0-1)	F1 (%)	Th	mIoU (0-1)	F1 (%)
U-Net (our)	0.75	**0.9170**	**94.03**	**0.9083**	93.27	0.65	**0.9508**	95.07	0.3	**0.9448**	93.58	0.5	**0.9178**	90.89	0.5	**0.9298**	92.27
U-Net-W (our)	0.75	0.9103	93.80	0.9001	92.79	0.65	0.9465	94.57	0.25	0.9344	92.41	0.5	0.9057	89.20	0.35	0.9170	90.67
U-Net++ (our)	0.75	**0.9141**	**94.20**	0.9064	**93.85**	0.60	**0.9526**	**95.22**	0.2	**0.9472**	**94.51**	0.45	**0.9186**	90.87	0.45	**0.9294**	**92.82**
Lozej [18] (FT-10)	0.7	0.9113	93.62	**0.9088**	**93.33**	0.75	0.9488	94.61	0.40	0.9280	92.20	0.45	0.9048	89.77	0.4	0.9163	90.90
Lozej [18] (FT-ES)	0.8	0.9116	94.01	0.9065	93.30	0.65	0.9506	94.67	0.25	0.9356	93.04	0.45	0.9104	90.66	0.45	0.9236	91.83
Trok [24] (FT)	n/a	0.8509	88.14	0.8520	88.38	n/a	0.8298	89.52	n/a	0.7938	85.28	n/a	0.8524	85.28	n/a	0.8421	83.42
Wang [7]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	94.25	n/a	n/a	**91.78**	n/a	n/a	**93.05**
Wang [23]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	89.40	**94.30**	n/a	84.79	91.33	n/a	n/a	n/a
Bezerra [1] (FCN)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	94.42	n/a	n/a	n/a	n/a	n/a	88.20	n/a	n/a	83.03
Bezerra [1] (GAN)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	**95.38**	n/a	n/a	n/a	n/a	n/a	**91.42**	n/a	n/a	87.42
OSIRIS v 4.1	n/a	0.8598	87.17	0.8547	87.81	n/a	0.8851	87.78	n/a	0.8338	82.87	n/a	0.4327	20.97	n/a	0.4911	32.48
USIT (Wahet)	n/a	0.8730	87.10	0.8601	85.36	n/a	0.8083	81.62	n/a	0.7034	72.45	n/a	0.3448	20.94	n/a	0.4058	26.03

Annotations: Th: Threshold, EL: sample with eyelash occlusion, U-Net-W: U-Net with weight map, Trok (FT): Trokielewicz [24] (fine-tuned), FT-10: fine-tuned for 10 epoch, FT-ES: fine-tuned using early stopping. All methods are trained or tuned independently on the five training datasets and then evaluated on the respective testing set for fair comparisons. Bold values indicate the top two highest accuracy results for each metric across the different segmentation models.

(C) Cross-dataset performance evaluation: Generalization capability is a cornerstone of biometric recognition systems, ensuring that models can effectively adapt to and classify new, unseen data. To assess the generalization capability of the implemented models, we conducted cross-dataset testing. The models, fine-tuned on the composite subset dataset, were tested on the CASIA Distance dataset without any further tuning. The CASIA Distance dataset was not included in our composite or composite subset dataset. Its images were captured from a three-meter distance under moving conditions with NIR illumination, making it distinct from the other CASIA datasets and unseen by the models trained on the composite or composite subset dataset. The cross-dataset performance is shown in Table 5.

Table 5. Cross-dataset performance (Test DB: CASIA Distance).

Method	Train DB	Th	mIoU	F1 Score (%)
U-Net (ours)	Composite	0.6	0.9231	91.43
U-Net++ (ours)	Composite	0.75	**0.9331**	**92.01**
U-Net with weight map (ours)	Composite	0.75	0.9150	89.72
Lozej [18] (original)	CASIA Interval	0.80	0.4347	6.67
Lozej [18] (fine-tuned)	Composite	0.65	0.9080	89.72
Trokielewicz [24] (original)	Postmortem	n/a	0.7398	79.28
Trokielewicz [24] (fine-tuned)	Composite	n/a	0.8137	87.91

Note: Bold values highlight the best accuracy results across the models for each metric.

Table 5 indicates that all of our proposed models exhibited strong generalization capabilities and outperformed all the implemented methods. The U-Net model achieved a mIoU of 0.9231 and an F1 score of 91.43%, while the U-Net++ model showed a slight improvement with a mIoU of 0.9331 and an F1 score of 92.01%. These results establish a standard benchmark for generalization capability in iris segmentation models. Conversely, the original model from Lozej et al. showed limited adaptability, with a mIoU of 0.4347 and an F1 score of 6.67% when applied to the CASIA Distance dataset. However, once fine-tuned on our composite subset dataset, the same model's performance was significantly enhanced, reaching a mIoU of 0.9080 and an F1 score of 89.72%. This demonstrates that even models initially lacking in generalization can achieve benchmark standards with the appropriate training dataset. Similarly, Trokielewicz et al.'s original model was outperformed by our benchmark models when tested on the CASIA Distance dataset. It performed with a mIoU of 0.7398 and an F1 score of 79.28%. After fine-tuning, there was a notable increase to a mIoU of 0.8137 and an F1 score of 87.91%, though this was still below the performance of our benchmark models. Overall, all the current open-source DL segmentation models have limited generalization capability. These results underscore the importance of comprehensive and diverse training in the development of models that excel not only in familiar conditions but also maintain high accuracy when faced with new and challenging datasets. Our benchmark models serve as a robust standard for the future development of the field of iris segmentation, guiding researchers toward creating more adaptive and reliable biometric recognition systems.

(D) Assessment of practicability: The practicability of the DL models is evaluated in terms of the number of parameters, storage space, and inference time (the duration required for a single prediction) of all models, as summarized in Table 6. These factors are critical for the deployment of models in practical applications, where efficiency and minimal resource consumption are often as important as accuracy. We measured the average inference time on a desktop configured with an Intel Core i9-12900K CPU, evaluating .h5 models. Our evaluation indicates the Wang et al. model requires the lowest storage space at 100 Mb, while Lojez's model requires 13 times more space at 13.8 GB. The Trokielewicz model has the fastest inference time at 0.004 s with a storage space of 104 Mb. In contrast, Lojez's model requires the highest inference time of 0.294 s. Our U-Net and U-Net with weighted map models have comparable storage space (360 MB) and inference time (0.02 s). Our U-Net++ model requires a storage space of 414 Mb and an inference time of 0.269 s. Overall, all models present significant computational and storage demands, making their implementation on mobile devices challenging. Therefore, further optimization of these models is required.

Table 6. An overview of deep learning models assessed in this study.

Research	Architecture	Input Image Resolution	Resource Availability		Eyelash Detection	No of Parameter	Storage Space	Average Inference Time
			Code	Ground Truth				
U-Net (ours)	U-Net	256 × 256	✓	✓	✓	31.46 M	360 MB	0.021 s
U-Net++ (ours)	U-Net++	256 × 256	✓	✓	✓	36.16 M	414 MB	0.269 s
U-Net-W (ours)	U-Net	256 × 256	✓	✓	✓	31.46 M	360 MB	0.024 s
Lozej et al. [18]	U-Net	320 × 320	✓		✓	124.36 M	1.38 GB	0.294 s
Trokielewicz et al. [24]	Seg-Net	160 × 120	✓	✓		29.4 M	104.65 MB	0.004s
Wang et al. [7]	Multi Tasking U-Net	n/a	✓	✓	✓	31.28 M	119 MB	n/a
Wang et al. [23]	Multi Tasking U-Net	n/a		✓	✓	n/a	100 MB	n/a
Bezerra [1]	FCN and GAN	512 × 512		✓	✓	n/a	n/a	n/a

(E) Visual evaluation: Figure 5 shows the segmentation results of the implemented methods from different datasets. Generally, noise such as eyelashes, eyelid occlusion,

specular reflections, uneven illumination, off-angle iris, and smaller-sized iris make the segmentation task difficult. To evaluate all models' performance on those edge cases, we visually assessed all challenging iris images from all the datasets. The findings are summarized here:

- DL methods showed improved noise masking capability compared with the traditional methods (OSIRIS and USIT) for cases like eyelid and eyelash occlusions, high and low illuminated samples, variable pupil dilation, smaller iris area, eyeglasses, and off-angle iris.
- A few incorrect segmentations were observed for cases with eyeglasses, eyelash occlusion, and off-angle iris samples. Limited training samples representing these cases might be the cause of these errors. Hence, adding more annotated training images with those cases may improve the segmentation performance.
- OSIRIS showed limited capability in eyeglass and eyelash detection.
- USIT could not detect eyelashes.
- Both OSIRIS and USIT performed poorly against off-angle iris.

Note: For visual comparison, we only compared OSIRIS and USIT's performance on the NIR dataset, as they were unsuitable for VIS datasets.

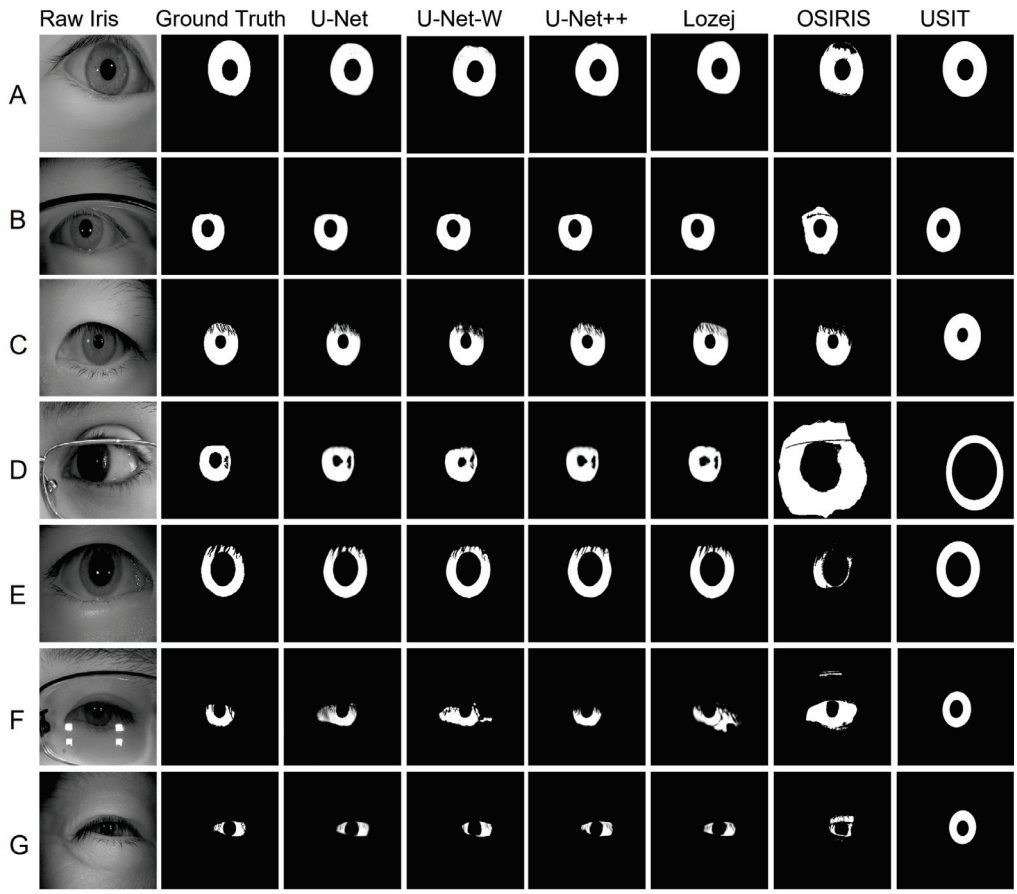


Figure 5. Examples of segmentation outputs from different models across various cases: (**A**) Iris without occlusions, (**B**) Iris with eyeglass occlusion, (**C**) Iris with eyelash occlusion, (**D**) Off-angle iris with eyeglass occlusion and specular reflection, (**E**) Eyelash occlusion with a dilated pupil, (**F**) Eyeglass occlusion and specular reflection, (**G**) Smaller iris area.

5. Discussion and Conclusions

In this study, we conducted a comprehensive review and benchmarked eleven iris segmentation models across five different datasets (including NIR and VIS) to evaluate their effectiveness and limitations. In the process, we prepared an end-to-end iris segmentation

resource set comprising a large training dataset (gleaned from multiple public sources), developed an iris segmentation toolkit, and manually annotated 1k images to address the unavailability of iris masks with eyelash labels. We developed three and adapted eight algorithms for comparison (two traditional and six DL-based).

Though NIR is predominantly utilized for iris recognition, there is a persistent interest [3,4] in the community to enhance VIS-based iris recognition for their widespread adaptability and less stringent hardware requirements. To assess the viability of VIS-based recognition, we incorporated two VIS-based datasets into our evaluation. Our evaluation demonstrates that DL models can adapt to VIS-based datasets and achieve comparable performance to NIR-based datasets upon fine-tuning, while traditional methods fail under these conditions. These efforts in research are driven by the desire to broaden the applications of VIS-based iris recognition, particularly for its potential to reduce hardware complexities associated with NIR collection, thus facilitating integration into portable devices like smartphones.

Our experimental findings demonstrate that Bezerra's FCN models perform well in the NIR dataset while showing less effectiveness in VIS datasets, especially in the MICHE-I dataset. This discrepancy is likely due to the greater variability in visible light images, which can include differences in lighting conditions, colors, and contrasts that are not as prevalent in NIR images. This suggests the need for improved models or training strategies that can better handle the diversity in VIS data. Similarly, the mixed success of Bezerra's GAN-based models across datasets underscores the requirement of careful adjustment to their architecture or training protocols to enhance their performance across a diverse range of real-world scenarios. Additionally, our analysis highlighted the consistent performance of all U-Net architecture-based models (developed by us, Lojez, and Wang) across both NIR and VIS datasets, demonstrating their flexibility under varying imaging conditions. However, the observation of marginal accuracy improvements among these DL models within specific datasets indicates a point of diminishing returns in focusing solely on architectural advancements. This suggests that augmenting the variety and representativeness of training datasets might offer more significant benefits, encouraging models to better capture and respond to the complexities encountered in practical applications.

The comparison with traditional approaches demonstrates that deep learning (DL) models show improvement over traditional methods for challenging samples, such as eyelid and eyelash occlusions, eyeglasses, high and low illuminated samples, variable pupil dilation, smaller iris areas, and off-angle irises. However, there is still scope for improvement in eyelash detection, eyeglasses, and off-angle iris segmentation. The consistent outperformance of DL models over traditional methods across all metrics and datasets reaffirms the transformative impact of DL in biometric recognition. However, Trokielewicz et al.'s [24] SegNet-based model, which is efficient in memory and inference time and designed for postmortem iris segmentation, demonstrates the trade-off between specialization and generalization. Its focus on postmortem data leads to reduced effectiveness on live samples while finetuning on live iris datasets, with accuracy dropping by 6–15% in mIoU and 6–9% in F1 scores compared with other DL approaches (Table 4). This underscores the importance of developing versatile models that maintain high performance across both live and specialized scenarios, highlighting the need for a balance between model specialization and general applicability in biometric recognition.

The proposed U-Net benchmark architecture is a balanced model showing comparable or better performance with high cross-dataset performance (only outperformed by our U-Net++) high eyelash detection capability (only surpassed by Lozej) with an inference time of 0.02 s (bested only by Trokielewicz) and a storage space of 360 Mb. Lozej's model shows comparable performance to our proposed U-Net model in certain datasets, but it falls short in generalization capability and is significantly less efficient, requiring 13 times more storage and a 14 times higher inference time.

Increasing the weight of the eyelashes to train the U-Net with the weight map model did not improve the accuracy. The dense convolution blocks in the skip connection of U-Net++ increased the computation and hardware cost of the network. It showed comparable overall performance with the U-Net (refer Table 4). However, the impact is seen in our generalizability assessment, where it outperforms all models by a considerable margin.

Wang et al. [7] performed an ablation study and showed multi-tasking U-Net (trained on the iris mask, pupil mask, and iris outer boundary mask) performed better than their U-Net (trained on only the iris mask). Adding an attention module with ASPP (Atrous Spatial Pyramid Pooling) improved the model's performance (0.23% on Miche-I and 0.89% on UBIRIS) compared with our models. However, annotation of the pupil mask and iris outer boundary mask needed more effort. Incorporating attention modules with ASPP (Atrous Spatial Pyramid Pooling), as proposed by Wang et al. [7], into our framework could potentially enhance our model's accuracy by capturing multi-scale contextual information and focusing on the most important features while suppressing irrelevant noise. ASPP, integrated within the attention module, enables the model to process both local and long-range spatial features. This may improve segmentation performance, particularly in challenging cases involving occlusions from eyelashes, eyeglasses, or other distractions. Additionally, the integration of ASPP within the attention mechanism could help address the scale-sensitivity problem, potentially allowing the model to detect iris features across varying scales, including images with different iris sizes and pupil dilation.

Our benchmark models, achieving high mIoU and F1 scores on the cross dataset (Table 5), set a new standard in the field, offering a robust framework for evaluating the generalization capabilities of future models. Current open-source models exhibit limited generalization capabilities and lag behind our benchmarked standards even after finetuning on our composite subset datasets (Table 5). This sets a challenge for the community to advance model performance and develop more adaptable, reliable biometric systems.

DL models often require substantial computational and storage resources. It's essential to develop more efficient DL networks to ensure their practicality for widespread use. Therefore, further optimization of existing benchmark models remains an important area for future research. Additionally, future research should address gaps in eyelash detection eyeglasses and off-angle iris segmentation, considering the trade-offs between model complexity and data availability. This evaluation showcases the critical role of training data diversity and model generalization, with models like Lozej et al. showing significant improvement after fine-tuning with diverse data. This underscores the need for innovative architectures and training datasets that are diverse and representative. A limitation of our research is the relatively small size of the fine-tuning datasets, each containing around 1000 samples. This was primarily due to the scarcity of annotated iris images with eyelash labels. Our open-source benchmark models, iris segmentation toolkit, and a manually annotated dataset of 1K iris masks will support and encourage further research in this field.

Author Contributions: Conceptualization, P.D., M.R.S., S.D., and S.S.; methodology, M.R.S. and P.D.; software, M.R.S., P.D., and A.H.; validation, M.R.S., P.D., and A.H.; investigation, M.R.S. and P.D.; data curation, M.R.S. and P.D.; writing—original draft preparation, M.R.S. and P.D.; writing—review and editing, M.R.S., P.D., S.D., and S.S.; supervision, S.D. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation under Grant No. 1650503.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The required datasets for the experiment can be obtained from: 1. Clarkson: https://tinyurl.com/43euw7ca (accessed on 24 February 2024). 2. CASIA: http://www.idealtest.org/dbDetailForUser.do?id=4 (accessed on 24 February 2024). 3. MICHE-I Email: biplab@unisa.it. 4. UBIRIS.v2 Email: hugomcp@di.ubi.pt.

Conflicts of Interest: The authors declare no conflicts of interest.

Glossary

Term	Definition
Iris segmentation	The process of isolating the iris from the rest of the eye in an image, including separating it from other structures like the eyelid, eyelashes, and surrounding areas.
Deep Learning (DL)	A subset of machine learning involving neural networks with multiple layers, used for tasks such as image segmentation and classification.
U-Net/U-Net++	Specific deep learning architectures designed for image segmentation tasks. U-Net++ enhances U-Net by adding nested convolutional blocks to improve feature representation.
Cross-dataset performance analysis	Evaluating a model's ability to perform well on datasets that were not used during its training phase, important for testing model generalization.
Generalization capability	The ability of a model to apply what it has learned from one dataset to new, unseen data from different datasets.
mIoU (Mean Intersection Over Union)	A metric used to evaluate the accuracy of segmentation models by comparing the overlap between predicted and ground truth areas.
F1 Score	A measure of a model's accuracy that considers both precision and recall, especially in binary classification tasks like pixel-wise segmentation.
NIR (Near-Infrared)	A part of the light spectrum used in iris imaging, typically in the wavelength range of 700 to 900 nanometers (nm), for capturing iris patterns under controlled lighting conditions.
VIS (Visible Spectrum)	The portion of the electromagnetic spectrum visible to the human eye, ranging from 380 to 720 nanometers (nm). While the use of VIS for iris recognition is still an area of ongoing research, it holds potential to reduce hardware complexities associated with NIR collection, making it easier to integrate iris recognition into portable devices such as smartphones.
Benchmarking	The process of comparing different models or algorithms using specific metrics across common datasets to evaluate their performance.
Eyeglash occlusion	A condition where parts of the iris image are blocked by eyeglashes, making segmentation more difficult.
Practicability	A measure of how feasible it is to implement a model in real-world applications, considering factors like inference time and computational requirements.
Inference time	The time a model takes to make a prediction, critical for real-time applications.
Ground truth	Manually labeled data used as a standard or reference for evaluating the performance of algorithms, such as manually segmented iris masks.
OSIRIS	An open-source iris recognition software that provides a complete iris recognition pipeline, including segmentation, feature extraction, and matching.
USIT	An iris recognition software that provides a complete open-source framework for iris segmentation, feature extraction, and matching, often used in biometric research alongside OSIRIS.

Composite dataset	A dataset created by combining samples from multiple publicly available datasets to form a larger, more diverse set of images for training or evaluation.
Noise detection capability	The ability of a segmentation algorithm to identify and handle artifacts like reflections or occlusions that can interfere with the accuracy of the iris segmentation process.
Specular reflection	The bright spot or glare in an image caused by light reflecting off curved surfaces, such as the eye or eyeglasses. This reflection can obscure iris details and negatively impact segmentation accuracy.
Dilation	The process by which the pupil expands in response to low light or other stimuli. Pupil dilation can alter the visible area of the iris, potentially impacting segmentation accuracy.

References

1. Bezerra, C.S.; Laroca, R.; Lucio, D.R.; Severo, E.; Oliveira, L.F.; Britto, A.S.; Menotti, D. Robust iris segmentation based on fully convolutional networks and generative adversarial networks. In Proceedings of the 2018 31st IEEE SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Parana, Brazil, 29 October–1 November 2018; pp. 281–288.
2. Kerrigan, D.; Trokielewicz, M.; Czajka, A.; Bowyer, K.W. Iris recognition with image segmentation employing retrained off-the-shelf deep neural networks. In Proceedings of the 2019 IEEE International Conference on Biometrics (ICB), Crete, Greece, 4–7 June 2019; pp. 1–7.
3. Song, Y.; He, Y.; Zhang, J. A survey of visible iris recognition. In Proceedings of the CS & IT Conference, Sydney, Australia, 23 March 2019; Volume 9.
4. Jenadeleh, M.; Pedersen, M.; Saupe, D. Blind quality assessment of iris images acquired in visible light for biometric recognition. *Sensors* **2020**, *20*, 1308. [CrossRef] [PubMed]
5. Daugman, J. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 1148–1161. [CrossRef]
6. Wildes, R. Iris recognition: An emerging biometric technology. *Proc. IEEE* **1997**, *85*, 1348–1363. [CrossRef]
7. Wang, C.; Muhammad, J.; Wang, Y.; He, Z.; Sun, Z. Towards complete and accurate iris segmentation using deep multi-task attention network for non-cooperative iris recognition. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2944–2959. [CrossRef]
8. Huo, G.; Lin, D.; Yuan, M. Iris segmentation method based on improved UNet++. *Multimed. Tools Appl.* **2022**, *81*, 41249–41269. [CrossRef]
9. Hofbauer, H.; Alonso-Fernandez, F.; Wild, P.; Bigun, J.; Uhl, A. A Ground Truth for Iris Segmentation. In Proceedings of the 2014 22nd International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 527–532.
10. IRISSEG-CC Dataset. Available online: http://islab.hh.se/mediawiki/index.php/Iris_Segmentation_Groundtruth (accessed on 24 February 2024).
11. IRISSEG-EP Dataset. Available online: http://www.wavelab.at/sources (accessed on 24 February 2024).
12. Othman, N.; Dorizzi, B.; Garcia-Salicetti, S. OSIRIS: An open source iris recognition software. *Pattern Recognit. Lett.* **2016**, *82*, 124–131. [CrossRef]
13. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
14. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Trans. Med. Imaging* **2020**, *39*, 1856–1867. [CrossRef] [PubMed]
15. Miron, C.; Pasarica, A.; Manta, V.; Timofte, R. Efficient and robust eye images iris segmentation using a lightweight U-net convolutional network. *Multimed. Tools Appl.* **2022**, *81*, 14961–14977. [CrossRef]
16. Wang, Q.; Meng, X.; Sun, T.; Zhang, X. A light iris segmentation network. *Vis. Comput.* **2022** , *38*, 2591–2601. [CrossRef]
17. Meng, Y.; Bao, T. Towards More Accurate and Complete Heterogeneous Iris Segmentation Using a Hybrid Deep Learning Approach. *J. Imaging* **2022**, *8*, 246. [CrossRef] [PubMed]
18. Lozej, J.; Meden, B.; Struc, V.; Peer, P. End-to-end iris segmentation using u-net. In Proceedings of the 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), San Carlos, Costa Rica, 18–20 July 2018; pp. 1–6.
19. Zhang, W.; Lu, X.; Gu, Y.; Liu, Y.; Meng, X.; Li, J. A robust iris segmentation scheme based on improved U-net. *IEEE Access* **2019**, *7*, 85082–85089. [CrossRef]
20. Jalilian, E.; Uhl, A. Iris segmentation using fully convolutional encoder–decoder networks. In *Deep Learning for Biometrics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 133–155.
21. Li, Y.H.; Putri, W.R.; Aslam, M.S.; Chang, C.C. Robust iris segmentation algorithm in non-cooperative environments using interleaved residual U-Net. *Sensors* **2021**, *21*, 1434. [CrossRef] [PubMed]

22. Chen, Y.; Wang, W.; Zeng, Z.; Wang, Y. An adaptive CNNs technology for robust iris segmentation. *IEEE Access* **2019**, *7*, 64517–64532. [CrossRef]
23. Wang, C.; Zhu, Y.; Liu, Y.; He, R.; Sun, Z. Joint iris segmentation and localization using deep multi-task learning framework. *arXiv* **2019**, arXiv:1901.11195.
24. Trokielewicz, M.; Czajka, A. Data-driven segmentation of post-mortem iris images. In Proceedings of the 2018 IEEE International Workshop on Biometrics and Forensics (IWBF), Sassari, Italy, 7–8 June 2018; pp. 1–7.
25. B. I. Test. Casia.v4 Database. Available online: http://www.idealtest.org/dbDetailForUser.do?id=4 (accessed on 22 February 2024).
26. Sequeira, A.; Chen, L.; Wild, P.; Ferryman, J.; Alonso-Fernandez, F.; Raja, K.B.; Raghavendra, R.; Busch, C.; Bigun, J. Cross-eyed-cross-spectral iris/periocular recognition database and competition. In Proceedings of the 2016 IEEE International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 21–23 September 2016; pp. 1–5.
27. De Marsico, M.; Nappi, M.; Riccio, D.; Wechsler, H. Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols. *Pattern Recognit. Lett.* **2015**, *57*, 17–23. [CrossRef]
28. Proença, H.; Filipe, S.; Santos, R.; Oliveira, J.; Alexandre, L.A. The UBIRIS.v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1529–1535. [CrossRef] [PubMed]
29. Warsaw-BioBase-Post-118 Mortem-Iris v1.0. Available online: http://zbum.ia.pw.edu.pl/EN/node/46 (accessed on 18 February 2024).
30. Kumar, A.; Passi, A. Comparison and combination of iris matchers for reliable personal authentication. *Pattern Recognit.* **2010**, *43*, 1016–1026. [CrossRef]
31. Lian, S.; Luo, Z.; Zhong, Z.; Lin, X.; Su, S.; Li, S. Attention guided U-Net for accurate iris segmentation. *J. Vis. Commun. Image Represent.* **2018**, *56*, 296–304. [CrossRef]
32. Rathgeb, C.; Uhl, A.; Wild, P. *Iris Recognition: From Segmentation to Template Security*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
33. Multimodal Biometric Dataset, Clarkson University. Available online: https://tinyurl.com/43euw7ca (accessed on 23 February 2024).
34. Verieye. Available online: https://www.neurotechnology.com/verieye.html (accessed on 20 February 2024).
35. Proença, H.; Alexandre, L.A. The nice. i: Noisy iris challenge evaluation—Part i. In Proceedings of the 2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems, Crystal City, VA, USA, 27–29 September 2007.
36. Hu, Y.; Sirlantzis, K.; Howells, G. Improving colour iris segmentation using a model selection technique. *Pattern Recognit. Lett.* **2015**, *57*, 24–32. [CrossRef]
37. Liu, N.; Li, H.; Zhang, M.; Liu, J.; Sun, Z.; Tan, T. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In Proceedings of the 2016 IEEE International Conference on Biometrics (ICB), Halmstad, Sweden, 13–16 June 2016; pp. 1–8.
38. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
39. Uhl, A.; Wild, P. Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation. In Proceedings of the 2012 IEEE 5th IAPR International Conference on Biometrics (ICB), New Delhi, India, 29 March–1 April 2012; pp. 283–290.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Detection of Human Traffic Controllers Wearing Construction Workwear via Synthetic Data Generation

Seunghyun Baik and Euntai Kim *

School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Republic of Korea; shbaik104@yonsei.ac.kr

* Correspondence: etkim@yonsei.ac.kr; Tel.: +82-2-2123-7729

Abstract: Developing Level 3 or higher autonomous vehicles requires the ability to follow human traffic controllers in situations where regular traffic signals are unavailable, such as during construction. However, detecting human traffic controllers at construction sites is challenging due to the lack of dedicated datasets and variations in their appearance. This paper proposes a method for detecting human traffic controllers by generating synthetic images with diffusion models. We introduce a color-boosting technique to enhance image diversity and employ a cut-and-paste mechanism for seamless integration into realistic road scenes. We generate 19,840 synthetic images, combined with 600 real-world images, to train a YOLOv7 model. The trained model achieves an AP_{50} score of 73.9%, improving by 32.9% over the baseline. The HTC600 dataset used in our experiments is publicly available to support autonomous driving research.

Keywords: human traffic controller detection; image synthesis; style and pose customization

1. Introduction

Autonomous vehicles are advancing to Level 3 and beyond. As they progress, they face increasingly complex and unpredictable road conditions, including construction zones. In these scenarios, vehicles cannot rely solely on static traffic signs or digital maps. They must detect human traffic controllers and interpret their gestures to navigate safely, as these controllers play a vital role in managing traffic flow. Failure to detect them can cause serious accidents. People might think that with recent advances in object detection, such as YOLO [1] and DETR-based approaches [2,3], detecting human traffic controllers would be straightforward. However, this is not the case. The main challenge in training human traffic controller detectors is the lack of dedicated datasets. Addressing this challenge is complicated by the diverse appearances of human traffic controllers. These include variations in workwear and helmet colors, as well as a wide range of signaling poses. Overcoming these challenges requires innovative approaches to data generation and augmentation.

Recent advancements in synthetic data generation have demonstrated their potential to tackle such data scarcity issues across various domains. Leveraging generative AI techniques, such as diffusion-based methods and AI-generated content, researchers have achieved significant improvements in object detection by creating diverse and realistic datasets [4–8]. Inspired by these advancements, this research aims to synthetically generate high-quality images of human traffic controllers with diverse workwear and poses. To address this problem specifically, we build on the synthetic data generation techniques that have been proven valuable in other domains [9–12] where synthetic data generation and enhancement techniques have been proven valuable.

The overall process consists of two main stages, as shown in Figure 1. Before starting these stages, we collect 600 real images of human traffic controllers and build a dataset named HTC600 (Human Traffic Controller dataset with 600 images) (A.1). In the first stage, we train a person image synthesis network, PIDM [13], using HTC600 and the DeepFashion dataset [14] (B.2). To further enhance the model's ability to produce diverse outputs, we produce a technique called color boosting for image synthesis (B.1). This technique diversifies the colors of workwear items such as helmets, vests, and pants, ensuring that the generated images effectively capture the wide range of visual characteristics associated with human traffic controllers.

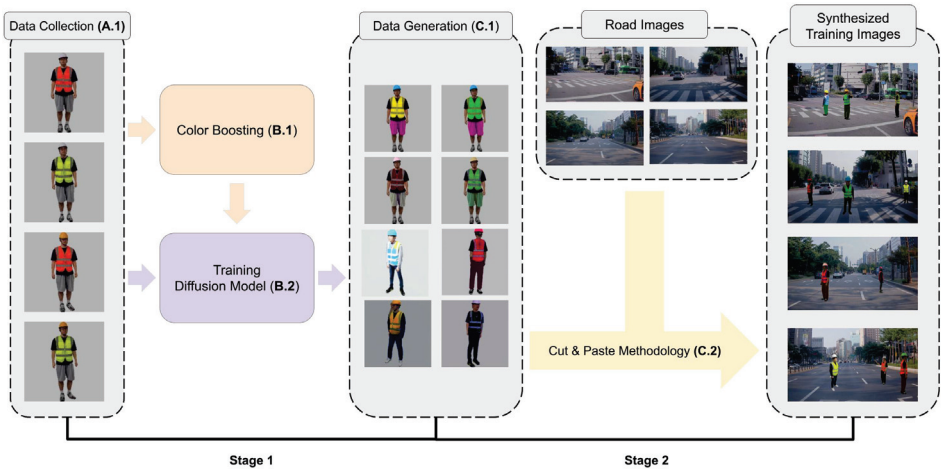


Figure 1. Overview of the proposed method: Generating diverse human traffic controller images from base images and integrating them with road images using a cut-and-paste mechanism to create realistic synthesized images.

In the second stage, we generate a synthetic human traffic controller using PIDM (C.1) and integrate the generated human traffic controller images into realistic road construction environments using a cut-and-paste methodology (C.2) inspired by prior works [9–12]. This approach places human traffic controllers in realistic positions and resizes them based on their scene placement, blending them seamlessly with the background. This step is critical for creating training images that not only include diverse human traffic controller appearances but also represent realistic scenarios in which human traffic controllers are encountered on roadways.

To evaluate the effectiveness of our approach, we built the HTC600 dataset, which consists of 600 real images for training and 135 real-world images for testing. The datasets include images captured in both controlled and uncontrolled environments, as well as images sourced from the Internet. This diverse dataset provides a robust foundation for benchmarking human traffic controller detection performance. The HTC600 dataset is publicly available for download at https://github.com/SeungHyun104/HTC600 (accessed on 16 December 2024).

In the experiment, we synthesized a large number of training datasets using only 600 human traffic controller images. This approach achieved a robust result with a AP_{50} score of 73.9% on the HTC600 test dataset.

In summary, this work makes the following contributions:

- We present the first study addressing human traffic controller detection through the generation of synthetic images.
- We propose a novel image synthesis framework that combines advanced techniques, such as color boosting and perspective-aware cut-and-paste integration, to generate

diverse and realistic datasets from limited initial data, specifically tailored for robust human traffic controller detection.
- We develop and publicly release the HTC600 dataset, comprising both real-world and synthetic images, to address data scarcity and support further research in human traffic controller detection.
- By leveraging the synthesized dataset, we achieve a significant improvement in detection performance, as evidenced by an increase in AP_{50} of 32.9% compared to the baseline.

Our experimental results confirm that using the synthesized dataset substantially improves the robustness and accuracy of human traffic controller detection models. This advancement is a critical step towards enhancing the safety and reliability of autonomous driving systems as they navigate the complexities of real-world road construction environments.

2. Related Works

2.1. Worker Detection

Worker detection in construction sites is important, especially with the rise of automated safety systems and autonomous vehicles. These tasks are generally approached as part of object detection, with specific adaptations for construction environments. Traditional methods often use models like YOLO [1]. These models include features like enhanced feature extraction and attention mechanisms to manage visually challenging scenarios. For example, Son et al. [15] trained YOLOv4 using large public datasets like ImageNet [16] and Pascal VOC [17], along with a construction site dataset of 22,000 images, to achieve robust worker detection.

Recent advancements have focused on addressing challenges such as occlusion, overlapping individuals, and varied worker postures. Park et al. [18] introduced SOC-YOLO, an adaptation of YOLOv5, which incorporated improvements such as a distance intersection over union (DIoU) loss function and a weighted-triplet attention mechanism. These modifications significantly enhanced detection accuracy for partially occluded workers in crowded environments, proving valuable for safety monitoring in complex construction settings.

2.2. Pose-Guided Person Image Synthesis

Pose-guided person image synthesis has seen significant advancements, especially with the introduction of various models aimed at generating human images in desired poses. Early methods, like Ma et al. [19], took a coarse-to-fine approach but often faced issues with feature misalignment due to the simple concatenation of source images and poses. To improve this, Esser et al. [20] separated appearance and pose using a VAE-based framework, introducing skip connections for better alignment. Siarohin et al. [21] refined this approach by employing deformable skip connections to handle local spatial transformations. Meanwhile, Ren et al. [22] developed GFLA, which used flow fields to warp image patches and improve pose alignment. Models like ADGAN [23] and PISE [24] introduced texture encoding and parsing maps to enhance the quality of synthesized images.

More recently, Bhunia et al. [13] proposed a diffusion-based model called PIDM. This model departs from traditional GAN-based approaches.

Instead of attempting complex pose transformations in a single pass, PIDM breaks the process into simpler conditional denoising steps, preserving appearance details more effectively. Additionally, PIDM uses disentangled classifier-free guidance to ensure the output image closely matches both the source image's style and the target pose, even in complex scenarios.

In our work, we adopt PIDM for pose-guided person image synthesis, leveraging its diffusion-based approach to generate realistic images while maintaining high-quality appearance details, even under challenging pose variations.

2.3. Synthetic Data Generation for Object Detection

Creating synthetic training data is essential for addressing the limitations of real-world datasets. This is particularly important in object detection and other machine learning tasks. Synthetic data generation techniques allow researchers to create diverse datasets that would be challenging or costly to collect in real-world settings. Recent advancements have focused on domain-specific applications, leveraging methods that ensure both realism and variability in the generated datasets.

Recently, diffusion-based models and generative AI play key roles in this field. Reutov et al. [4] leveraged diffusion models to produce urban traffic datasets for vehicle detection, while Zhu et al. [5] introduced ODGEN for synthetic data generation in specialized object detection tasks, demonstrating substantial improvements in model robustness. Likewise, Chen et al. [6] developed RailFOD23 for detecting anomalies on railway transmission lines, showcasing how generative AI can boost detection accuracy.

In addition, other methods have proven effective in enhancing the adaptability of synthetic datasets across various domains. Oric et al. [7] added diverse environmental variables to synthetic vehicle detection datasets, and Goyal and Mahmoud [8] surveyed advanced techniques to highlight the importance of realism and variability. In construction-related detection, Seong et al. [11] introduced a content-swapping technique that decomposed and recombined sign components to mitigate data scarcity and improve model accuracy. Similarly, Lee et al. [12] developed an image synthesis framework leveraging domain transfer and blending techniques, effectively addressing data scarcity for fallen person detection and enhancing detection performance across diverse road scenarios.

Building on these advancements, our work adopts a cut-and-paste methodology but incorporates novel enhancements tailored to human traffic controller detection. Specifically, we integrate a color boosting method and the PIDM model [13] to generate a diverse and comprehensive synthetic dataset. The color boosting method utilizes Grounding DINO [25] and SAM2 [26] models to modify human traffic controller attire colors, creating variability while maintaining dataset integrity. The PIDM model further complements this by generating realistic images with varied poses and styles, addressing the challenges of pose diversity and appearance variability.

By combining these techniques, our approach creates a robust synthetic dataset that captures the complex visual characteristics of human traffic controllers in road construction environments. This dataset not only alleviates the challenges of collecting sufficient real-world data but also enhances the robustness and accuracy of detection models in scenarios where diverse appearances and poses of human traffic controllers are critical. Our work specifically addresses the unique challenges of detecting human traffic controllers in dynamic road conditions, bridging the gap left by traditional worker detection approaches.

2.4. Model Compression

As deep learning models continue to grow in size and complexity, they often face deployment challenges on resource-constrained devices. Model compression techniques, such as pruning and quantization, have emerged as effective solutions to reduce both computational demands and memory usage while still maintaining acceptable levels of performance.

Pruning involves removing unnecessary weights or channels to yield a more compact and efficient network structure. For instance, Fang et al. propose DepGraph [27], a

framework that generalizes structural pruning across various neural network architectures by analyzing dependencies between layers.

Quantization, on the other hand, reduces the numerical precision of weights and activations. This approach can significantly decrease model size and inference latency, particularly on specialized hardware. Günay et al. present LPYOLO, a low-precision variant of the YOLO framework tailored for face detection on FPGA devices [28]. Their experiments show that carefully quantized object detection networks can achieve near-real-time performance with a notable reduction in power consumption.

Overall, pruning and quantization are complementary strategies for model compression: pruning effectively removes redundant network components, while quantization optimizes numerical representation. Together, they allow deep models to run efficiently on hardware with limited resources, providing a promising path forward in scenarios where both latency and energy efficiency are paramount.

In our own work, we also incorporate quantization into YOLOv7 to further reduce computational overhead and facilitate real-time deployment on resource-constrained devices.

3. Methods

3.1. Overview

The two stages shown in Figure 1 are further divided into five key steps, as illustrated in Figure 2. These five steps can be summarized as follows:

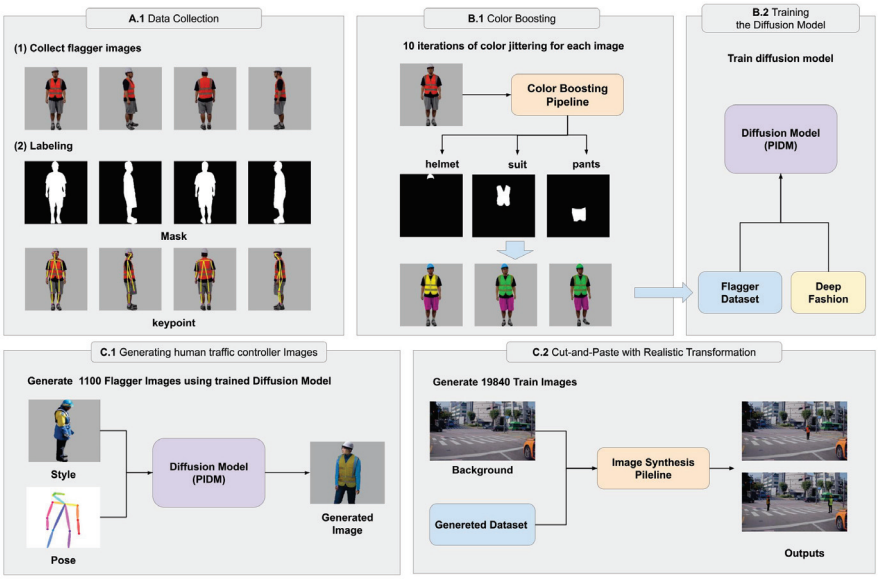


Figure 2. Detailed stage-by-stage overview of the proposed method: The entire process is divided into five steps. (**A.1**) Collecting human traffic controller images and annotating masks and keypoints. (**B.1**) Applying color boosting to diversify human traffic controller appearances. (**B.2**) Training the diffusion model with the augmented human traffic controller and DeepFashion datasets. (**C.1**) Generating diverse human traffic controller images with specified poses and styles using the trained diffusion model. (**C.2**) Integrating the generated images with road backgrounds using a cut-and-paste mechanism with realistic transformations to create a large synthesized training dataset.

- **Data Collection (A.1)**: We capture human traffic controller images featuring six different individuals. We took the images indoors from various angles, with subjects in a few fixed postures to maximize variability while using a small dataset. We annotated the images with keypoints and masks to support subsequent processing.
- **Color Boosting (B.1)**: A two-stage color boosting process applies to diversify the appearance of human traffic controller images. This process targets specific body

parts, such as helmets, suits, and pants, to introduce color variations while preserving dataset integrity.

- **Training the Diffusion Model (B.2)**: The diffusion model PIDM trains using the augmented human traffic controller dataset and the DeepFashion dataset. This enables the generation of realistic images with diverse poses and styles.
- **Generating Human Traffic Controller Images (C.1)**: Using the trained diffusion model, we synthesize a large set of high-quality human traffic controller images with specified poses and styles, ensuring broad variability in the generated dataset.
- **Cut-and-Paste with Realistic Transformations (C.2)**: Human traffic controller images seamlessly integrate into road background images using a cut-and-paste method. Realistic transformations, including perspective-based instance size adjustments, color harmonization, and multi-instance integration, apply to ensure the composites reflect real-world environments.

For a clearer explanation, we provide a pseudo-code of the proposed method in Algorithm 1. Each step in Algorithm 1 matches Figure 2. In the following subsections, we describe the details of each step.

Algorithm 1 Pseudo-code of the proposed method

A.1: Data Collection

Collect human traffic controller images I_{HTC}
Generate keypoints (K_{HTC}) and masks (M_{HTC}) of collected images.

B.1: Color Boosting

Generate masks of each body part (M_p, M_h, M_v) using Ground DINO and SAM2.
Sample brightness ($b_i \sim \mathcal{U}(0.5, 1.5)$), contrast ($c_i \sim \mathcal{U}(0.6, 1.4)$), saturation ($s_i \sim \mathcal{U}(0, 2.0)$), and hue ($h_i \sim \mathcal{U}(-0.4, 0.4)$).
Apply color adjustments:
$I'_i = I \cdot \text{AdjustBrightness}(b_i) \cdot \text{AdjustContrast}(c_i) \cdot \text{AdjustSaturation}(s_i) \cdot \text{AdjustHue}(h_i)$.
Combine adjusted regions and background:
$I_{\text{boosted}} = I'_p \odot M_p + I'_h \odot M_h + I'_v \odot M_v + I \odot (1 - (M_p + M_h + M_v))$.

B.2: Training the Diffusion Model

Train PIDM model with Collected and Color Boosted Datasets.

Input: $I_{\text{DeepFastion}}$, I_{boosted}, I_{HTC}, K_{HTC}

C.1: Generating Human Traffic Controller Images

Collect target style human traffic controller images S_{target}
Generate target keypoints K_{target}
Generate human traffic controller image $I_{\text{generated}}$ with target pose K_{target} and style K_{target}.

Input: S_{target}, K_{target}
Output: $I_{\text{generated}}$

C.2: Cut-and-Paste with Realistic Transformations

Collect road background images $I_{background}$
Identify pasteable regions $M_{pasteable}$ in $I_{background}$.
Compute vanishing point y_{max} and adjust human traffic controller size (H') according to where to attach y :
$H' = H_{\text{base}} \cdot \left(1 - \frac{y}{y_{\max}}\right)$.
Apply cut-and-paste to integrate I_{HTC}, I_{boosted}, $I_{\text{generated}}$, into $I_{background}$.
Harmonize colors between the pasted instance and background ($I_{background}$).

Output: Final synthesized dataset ($I_{\text{synthetic}}$).

3.2. Data Collection

As shown in Figure 3, we collected 600 images of human traffic controllers, featuring five different individuals. The five individuals wear four distinct combinations of white and yellow helmets with orange and yellow construction workwear suits. The images are taken from various angles, with individuals posed in a few fixed postures to efficiently generate diverse training datasets.

For labeling, we utilize pre-trained models to streamline the process. Specifically, we use the YOLOv7-pose model [29] for keypoint labeling and the Segment Anything 2 model [26] for mask labeling. This ensures accurate and efficient annotation of both keypoints and masks across the dataset, facilitating further analysis and model training.

Figure 3. The example of collected human traffic controller images.

3.3. Color Boosting

To enhance data variability, we develop a color boosting method consisting of two sub-steps, as illustrated in Figure 4. This approach targets specific body parts, such as the helmet, construction workwear suit, and pants, to introduce diverse color variations while maintaining dataset integrity.

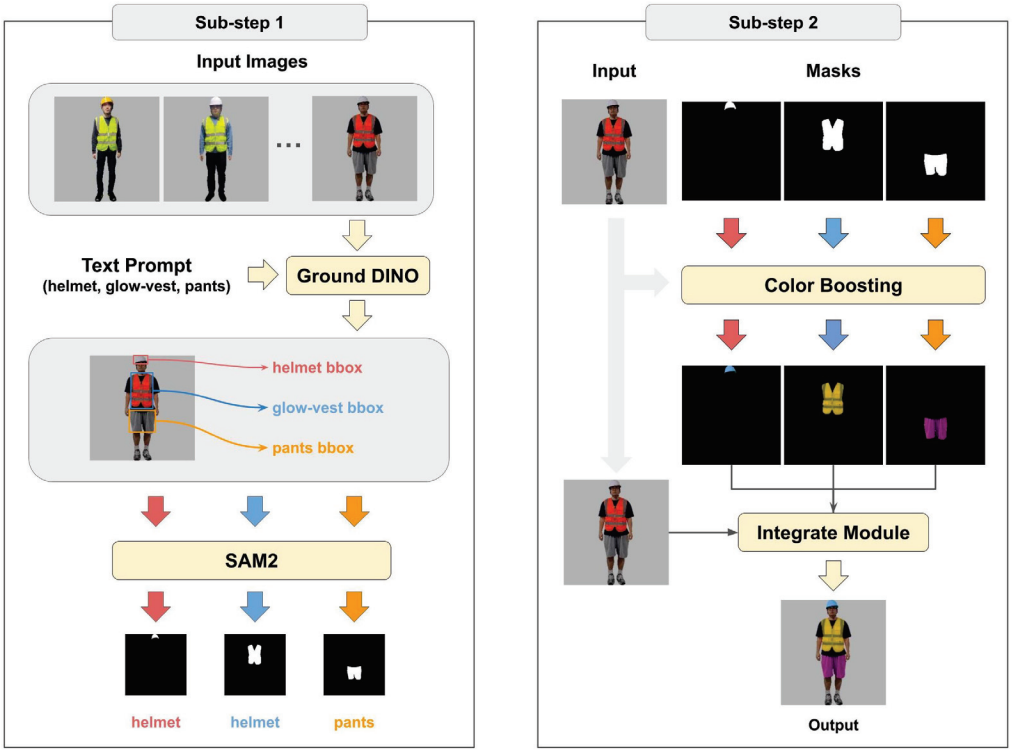


Figure 4. Color boosting process. In sub-step 1, Ground DINO and SAM2 generate masks for specific body parts. In sub-step 2, custom color boosting is applied to these regions to create diverse color variations.

3.3.1. Sub-Step 1: Mask Generation

We employ Ground DINO [25] and SAM2 [26] models to generate masks for each body part. Ground DINO uses text prompts to identify bounding boxes for the helmet, glow-vest, and pants, which are then passed to SAM2 to produce detailed masks. The mask for each region is defined as:

$$M_i(x,y) = \begin{cases} 1 & \text{if pixel } (x,y) \text{ belongs to region } i, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $i \in \{p, h, v\}$ represents the pants, helmet, and glow-vest regions, respectively.

3.3.2. Sub-Step 2: Color Boosting

To introduce diverse color variations, the brightness (b_i), contrast (c_i), saturation (s_i), and hue (h_i) for each region are independently adjusted. These attributes are sampled from uniform distributions as follows:

$$b_i \sim \mathcal{U}(0.5, 1.5), \qquad c_i \sim \mathcal{U}(0.6, 1.4), \qquad s_i \sim \mathcal{U}(0, 2.0), \qquad h_i \sim \mathcal{U}(-0.4, 0.4), \tag{2}$$

where $i \in \{p, h, v\}$ represents pants, helmet, and glow-vest regions. The color-adjusted regions are computed as:

$$I_i' = I \cdot \text{AdjustBrightness}(b_i) \cdot \text{AdjustContrast}(c_i) \cdot \text{AdjustSaturation}(s_i) \cdot \text{AdjustHue}(h_i). \tag{3}$$

To apply these adjustments to specific body parts, we use masks generated in Sub-step 1. Each body part is extracted using its corresponding mask:

$$I_p = I_p' \odot M_p, \quad I_h = I_h' \odot M_h, \quad I_v = I_v' \odot M_v, \quad I_{\text{bg}} = I \odot \big(1 - (M_p + M_h + M_v)\big), \tag{4}$$

where I_{bg} represents the background region excluding the masked parts.

Finally, the adjusted components are combined to produce the final image:

$$I_{\text{final}} = I_p + I_h + I_v + I_{\text{bg}}. \tag{5}$$

This process is repeated 10 times per image, introducing various variations in the appearances of human traffic controllers.

Figure 5 shows the transformation achieved by color boosting. The input image represents the original human traffic controller, while the output images demonstrate diverse variations in the color attributes of helmets, glow-vests, and pants. The corresponding mask images highlight the regions targeted for adjustment.

The method introduces realistic variability in brightness, contrast, saturation, and hue, ensuring that only the intended regions are modified. This targeted approach maintains the structural integrity of the original image while simulating diverse environmental conditions. By generating visually diverse samples, the color boosting process enhances the robustness of the dataset, enabling the detection model to better handle variations in workwear and lighting conditions encountered in real-world scenarios.

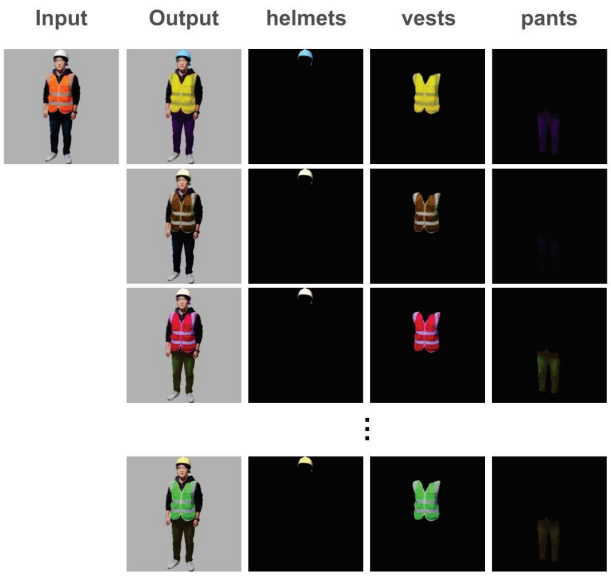


Figure 5. Examples of color boosted human traffic controller images. The generated masks for helmets, vests, and pants accurately correspond to the specified test prompts, as seen in the mask images. The color variations applied to each region demonstrate successful adjustments, showcasing diverse brightness, contrast, saturation, and hue changes while preserving the structural integrity of the original image.

3.4. Training the Diffusion Model

We train the PIDM [13] model using the collected and color-boosted human traffic controller images, combined with the DeepFashion dataset [14], to generate high-quality images with diverse poses and styles. The overall training process follows the methodology outlined in the original work [13].

The generative modeling process is based on the Denoising Diffusion Probabilistic Model (DDPM), which progressively adds Gaussian noise to the input data during the forward diffusion process, transforming it into pure noise. The forward process is defined as:

$$q(y_t|y_{t-1}) = \mathcal{N}(y_t; \sqrt{\alpha_t}y_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{6}$$

where y_t is the noisy data at time step t, α_t is a noise scheduling coefficient, and $\mathbf{I}$ is the identity matrix. At each step, Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is added, perturbing the original data y_0 into noise y_T over T timesteps. The resulting noisy data y_t at any step can be expressed directly as:

$$y_t = \sqrt{\bar{\alpha}_t}y_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{7}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ is the cumulative noise scheduling coefficient.

The reverse process reconstructs the original data by iteratively removing the added noise. It is parameterized as:

$$p_\theta(y_{t-1}|y_t, x_s, x_p) = \mathcal{N}(y_{t-1}; \mu_\theta(y_t, t, x_s, x_p), \Sigma_\theta(y_t, t, x_s, x_p)), \tag{8}$$

where x_s is the target style image, x_p is the target pose, and μ_θ, Σ_θ are the predicted mean and variance.

During training, the model learns to predict the added noise ϵ_θ at each step, minimizing the difference between the actual added noise ϵ and the predicted noise ϵ_θ. This is achieved by optimizing the simplified loss function:

$$L_{\text{mse}} = \mathbb{E}_{y_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(y_t, t, x_s, x_p)\|_2^2\right], \tag{9}$$

where ϵ is the actual noise added during the forward process, and ϵ_θ is the noise predicted by the model. This ensures accurate denoising during the reverse process.

To enhance the stability and quality of training, the hybrid loss function used in [13] combines this Mean Squared Error (MSE) loss with a variance improvement loss:

$$L_{\text{hybrid}} = L_{\text{mse}} + L_{\text{vib}}. \tag{10}$$

Here, L_{vib} improves the model's variance prediction Σ_θ, helping to stabilize the denoising process and refine the quality of the generated images.

By explicitly modeling the relationship between the added noise ϵ and the predicted noise ϵ_θ, the training process enables the model to reconstruct realistic images from noisy inputs. These iterative forward and reverse steps allow PIDM to achieve robust transformations, resulting in high-quality image generation with accurate poses and styles.

3.5. Generating Human Traffic Controller Images

Using the trained PIDM model, we generate human traffic controller images to create a robust and diverse dataset. For the generation process, we utilize 50 distinct poses and 22 different styles, resulting in a total of 1100 human traffic controller images (50 poses × 22 styles). These images were designed to include various combinations of styles and poses, ensuring a wide range of variability.

Figure 6 illustrates examples of human traffic controller images synthesized using the PIDM model. These images show the successful application of diverse styles and poses, capturing a wide range of variations in workwear and signaling gestures.

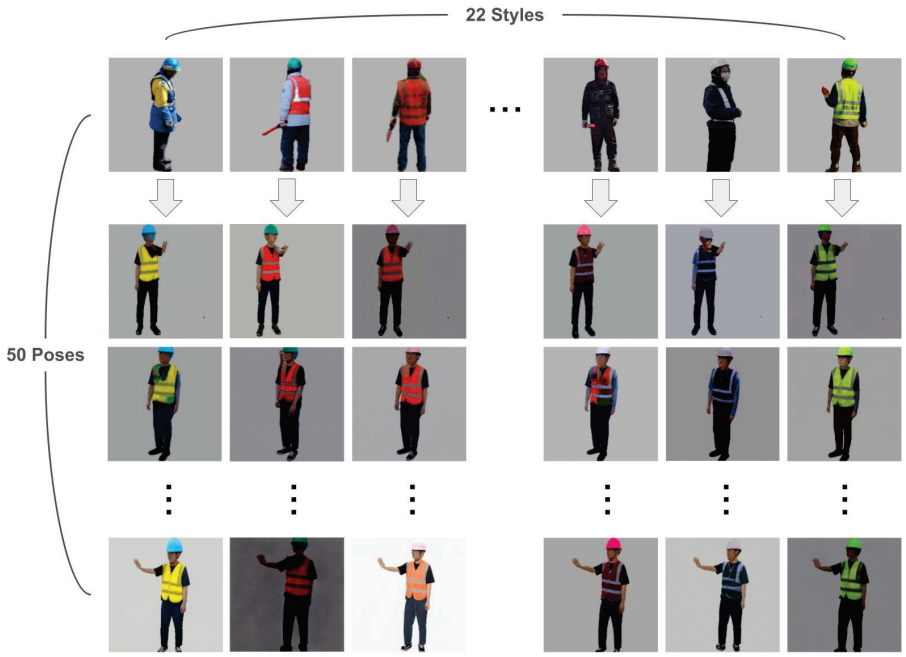


Figure 6. Examples of generated human traffic controller images. The human traffic controller images are created according to the desired styles and poses.

The ability to generate visually realistic and diverse images highlights the effectiveness of the PIDM model in preserving key visual details, such as helmet designs, glow-vest patterns, and body postures. By accurately synthesizing these characteristics, the model

significantly enriches the dataset and addresses the challenges posed by the limited availability of real-world data. This diverse data generation improves the robustness of the detection model by providing comprehensive examples that simulate real-world conditions. The synthesized images expand the model's capacity to handle variability in human traffic controller appearances, thereby improving detection performance in dynamic and complex environments.

3.6. Cut-and-Paste with Realistic Transformations

To synthesize realistic training datasets, we employ a cut-and-paste method [9,12,30,31] enhanced with realistic transformations. This process is further divided into three sub-steps: (1) identification of pasteable regions, (2) perspective-based instance size adjustments, (3) multi-instance integration and color adjustments. These sub-steps are designed to seamlessly integrate human traffic controller images into road backgrounds, ensuring that the synthesized images appear natural and realistic.

3.6.1. Pasteable Region Identification

This sub-step is based on our previous work [11]. We first identify pasteable regions as drivable areas in the road background images. These regions were detected using a pre-trained segmentation model, ensuring that human traffic controller images were placed only in feasible locations. The road background images are also taken from [11], which released 992 images collected in Seoul, South Korea.

3.6.2. Instance Size Adjustment Using the Vanishing Point

We develop an instance size adjustment method to paste a human traffic controller into a background image. This sub-step distinguishes the paste in this paper from the one in our previous work [11]. In that work, we used *simple image projection* to paste a road construction sign that appeared below the camera level, whereas, in this paper, we use the vanishing point to paste a human traffic controller shown at or above the camera level. The size adjustment method used here is inspired by [30,32].

Specifically, we use the vanishing point to adjust the size of each human traffic controller based on its vertical position (y) in the scene. The base height (H_{base}) of the human traffic controller is set to 750 pixels, representing the object's height when positioned at the bottom of the image ($y = 0$), but this value can vary depending on the background image height. The vanishing point ($y_{\max}$) corresponds to the biggest y-coordinate within the drivable region, as determined by a pre-trained segmentation model. For $0 < y < y_{\max}$, the size of the human traffic controller (H') is computed as:

$$H' = H_{\text{base}} \cdot \left(1 - \frac{y}{y_{\max}}\right), \quad 0 < y < y_{\max}, \tag{11}$$

where

- H_{base}: The base height of the human traffic controller image in pixels.
- y: The vertical position of the human traffic controller in the background image.
- $y_{\max}$: The largest y-coordinate within the drivable region, representing the vanishing point.

This equation ensures that the height of the human traffic controller decreases proportionally with its vertical position y, maintaining realism by scaling according to the perspective effect, while ensuring $H' > 0$.

This approach ensures that objects placed higher in the image appear smaller, accurately reflecting the perspective effect. Figure 7 illustrates this process, showing how the human traffic controller size changes based on its position relative to the vanishing point.

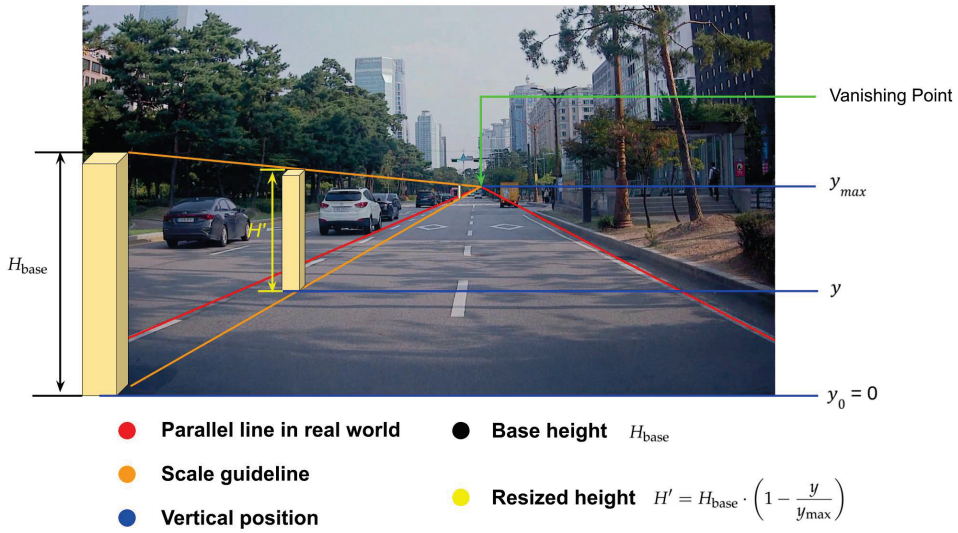


Figure 7. Illustration of the vanishing point-based instance size adjustment. The figure demonstrates how objects positioned higher in the image (closer to the vanishing point) are scaled down to reflect the real-world perspective. The base height (H_{base}) represents the object's original size at $y = 0$, and the resized height (H') adapts according to the vertical position (y) and maximum visible y-coordinate (y_{max}). Two red lines converge at the vanishing point, while scale guidelines in orange indicate the progressive reduction in object size.

3.6.3. Multi-Instance Integration and Color Adjustments

For each road background, we generate up to three instances of human traffic controllers per image, each subjected to independent size adjustment. This allows us to simulate more complex and realistic scenarios. By varying the number and position of human traffic controllers, we ensure a diverse set of training images.

In addition, to enhance the realism of the synthesized images, we apply color adjustment techniques that harmonize the appearance of the pasted human traffic controller instances with the background. This approach takes advantage of the color difference method introduced in our previous work [11]. By aligning the brightness, contrast, and overall tone of the human traffic controller images to match the road background, we minimized visual discrepancies and ensured seamless integration between the pasted objects and their surroundings.

3.6.4. Dataset Statistics

Using this approach, we synthesize a total of 19,840 training images per generation by combining human traffic controller instances with 992 road background images. For each road background, we generate 20 synthetic variations, ensuring a diverse and representative dataset. These realistic synthetic images significantly enhance the robustness of our detection model by providing a wide range of scenarios for training.

In particular, we mitigate the risk of overfitting through the methods outlined in Sections 3.3–3.6.3. First, the color boosting technique (Section 3.3) expands the color distribution of our training data, preventing the model from memorizing a single hue or brightness profile of human traffic controllers (HTCs). Second, *Generating human traffic controller images* (Section 3.5) via our diffusion-based approach introduces novel, previously unseen combinations of styles and poses. Third, the *pasteable region identification* (Section 3.6.1) restricts HTC placement to semantically valid areas, avoiding impractical placements that could bias the model. Fourth, *Instance Size Adjustment Using the Vanishing Point* (Section 3.6.2) incorporates perspective realism, preventing the model from overfitting to a single scale or viewpoint. Lastly, *multi-instance integration* (Section 3.6.3) produces

one to three HTCs per image, exposing the model to more complex relationships and reducing its reliance on any single scene composition.

As a result of these combined strategies, the dataset remains both diverse and true to real-world conditions, strengthening the model's generalization capabilities. In total, the HTC600 dataset comprises 600 real human traffic controller images, 6000 color-boosted real human traffic controller images, 1100 PIDM-generated images, the synthesized training datasets described above, and 135 real-world test images containing at least one human traffic controller instance.

4. Experiments

4.1. Implementation Details

We conducted experiments using our human traffic controller detection dataset, HTC600, to evaluate the effectiveness of the proposed methodology. We employed YOLOv7 [29] as the detection model. We trained the model on the HTC600 training datasets described in Section 3.6.4, each consisting of 19,840 synthesized training images generated by combining human traffic controller instances with 992 road background images. The key experimental settings and parameters are summarized in Table 1.

Table 1. Experimental details.

Parameter	Details
Model	YOLOv7
Training Dataset	HTC600 (19,840 synthesized images)
Input Resolution	640×640
Learning Rate	Initial: 10^{-2}, Final: 10^{-4} (cosine decay)
Training Epochs	200–300 (with early stopping)
Batch Size	32
Optimizer	Adam (lr: 10^{-2}, betas: (0.937, 0.999), weight decay: 5×10^{-4})
GPU	NVIDIA Titan RTX
Inference Settings	NMS (IoU: 0.45), Confidence Threshold: 0.25
Evaluation Metric	AP_{50}, AP (COCO Detection Benchmark)

For training, we used an RGB input resolution of 640×640. We initially set the learning rate at 10^{-2} and gradually decreased it to 10^{-4} using a cosine decay schedule. We trained the model for 200 to 300 epochs, with early stopping applied if the accuracy did not improve for more than 10 consecutive epochs. We used the Adam optimizer[33], with $\beta_1 = 0.937$ and $\beta_2 = 0.999$, and a weight decay of 5×10^{-4}. We set the batch size to 32, and the training process required approximately 60 h on a single NVIDIA Titan RTX GPU.

During inference, we applied Non-Maximal Suppression (NMS) with an Intersection over Union (IoU) threshold of 0.45 and a confidence threshold of 0.25 to refine predictions by eliminating overlapping detections. This process allowed the model to accurately detect human traffic controllers in various poses and environments while reducing noisy redundant results.

4.2. Quantitative Results

We evaluated the quantitative performance of the proposed method using the HTC600 test dataset, which contains 135 test images with at least one human traffic controller instance. For evaluation, we measured the Average Precision (AP) at a 0.5 IoU threshold (AP_{50}). AP_{50} represents the precision at a fixed IoU threshold of 0.5, while AP is calculated by averaging precision values at 10 IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05, following the COCO Detection Evaluation Benchmark https://cocodataset.org/#detection-eval (accessed on 16 December 2024).

To highlight the advantages of our approach, we established a baseline by applying a simple cut-and-paste method to generate 19,840 training images using the 600 collected human traffic controller images described in Section 3.1. Starting from this baseline (A), we incrementally applied the proposed enhancements, including realistic transformations (B), color boosting (C), use of PIDM-generated images (D), and color adjustment (E). Table 2 presents the experimental results for the HTC600 test dataset. As shown in Table 2, the baseline approach achieved an AP_{50} of 41.0%. By incorporating the proposed methods step-by-step, the performance significantly improved, reaching an AP_{50} of 73.9%. This represents a substantial improvement of 32.9%, demonstrating the effectiveness of our approach in enhancing human traffic controller detection accuracy. Each enhancement contributed meaningfully to the overall improvement, underscoring the robustness of the proposed method.

Table 2. YOLOv7 detection performance on HTC600 test dataset. (Units: Both AP_{50} and AP are expressed in %, calculated following the COCO detection evaluation benchmark.)

Method	$\boldsymbol{AP_{50}}$ **(%)**	$\boldsymbol{AP}$ **(%)**
A. Baseline (Cut and Paste)	41.0	19.6
B. + Pasteable Region and Instance Size	65.7	31.4
C. + Color Boosting	69.6	34.0
D. + Generated Images	72.9	39.2
E. + Color Adjustment	73.9	38.8

In Table 3, we conduct an ablation study using a state-of-the-art YOLO model. Specifically, we evaluate our method with YOLOv11n (Nano) and YOLOv11s (Small) [34]. The baseline represents method (A) from Table 2, while the proposed corresponds to the integration of methods (A) through (E). The results in Table 3 demonstrate that our proposed method significantly improves detection quality, highlighting its effectiveness in enhancing detection capabilities even when applied to advanced detection networks. The experiments were conducted using the same parameters as the original setup to ensure a fair comparison.

Table 3. Ablation study with YOLOV11 networks.

Method	**YOLOv11n (Nano)**		**YOLOv11s (Small)**	
	$\boldsymbol{AP_{50}}$ **(%)**	$\boldsymbol{AP}$ **(%)**	$\boldsymbol{AP_{50}}$ **(%)**	$\boldsymbol{AP}$ **(%)**
Baseline (A)	51.2	24.5	61.5	35.8
Proposed (A–E)	77.8	53.0	86.8	59.4

In Table 4, we present an experiment comparing our proposed method with the existing Fallen Person Detection (FPD) method [12]. FPD uses a combination of methods A and E from Table 2, while the proposed corresponds to the integration of methods (A) through (E). Using the FPD method, we generated a training dataset equivalent in size to the HTC600 training dataset, resulting in 19,840 synthetic training images. We conducted an experiment using the HTC600 test dataset, and the results demonstrate that our proposed method outperforms FPD, highlighting its effectiveness in enhancing detection capabilities under challenging conditions.

Table 4. Detection performance comparison on the HTC600 test dataset between FPD [12] and the proposed method.

Method	YOLOv7	
	AP_{50} (%)	AP (%)
FPD [12]	62.2	36.6
Proposed (A–E)	73.9	38.8

4.3. Qualitative Results

The total synthesized training datasets are shown in Figure 8. In the baseline method (A), the human traffic controllers are often placed in unrealistic locations, such as areas where they would not naturally appear in real-world scenarios. By incorporating pasteable region identification and instance size adjustments (B), we position the human traffic controllers on the road with more realistic scaling, improving visual coherence. Although the color boosting method (C) introduced a wider range of color variations for human traffic controller clothing, the limited number of original human traffic controller images still constrained dataset diversity. By integrating generated images (D), we were able to overcome this limitation by introducing a variety of poses and styles not present in the original dataset, thereby reducing the risk of overfitting. Finally, the color adjustment step (E) harmonized the synthesized human traffic controllers with the road background, creating more realistic composites and improving detection performance in domain-specific scenarios.

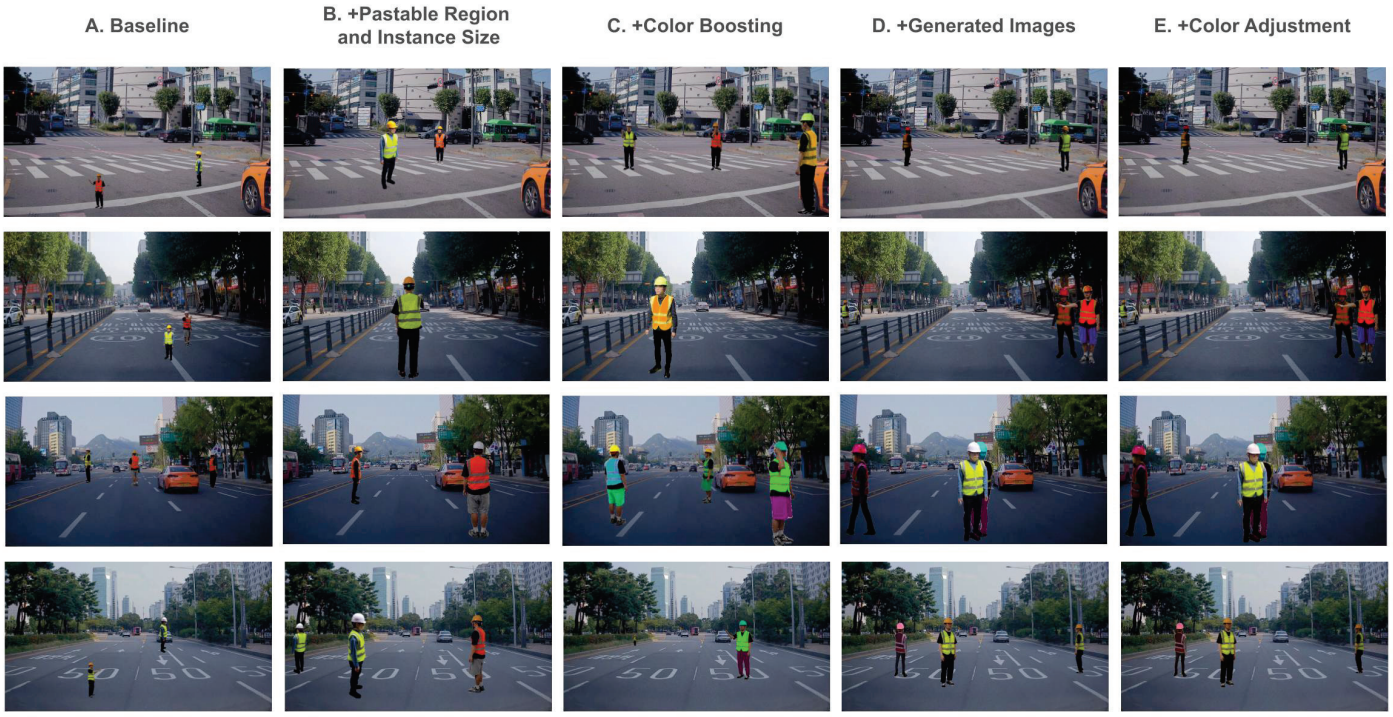


Figure 8. Synthesized training datasets. For each method, we sampled from the same set of background images.

Figure 9 illustrates the incremental improvements in detection accuracy when training models with methods (A) through (E). As more of our proposed components are added from the left column to the right column, we observe a clear reduction in incorrectly detected results (red) and an increase in correctly detected results (green), highlighting the model's growing generalizability.

Methods (A) and (B) provide moderate improvements by refining where and how large the human traffic controllers are placed. Method (C) allows the model to detect new styles that were not in the original HTC600 dataset. Then, method (D) enables the model to find poses it missed before, as shown in the second row where two human traffic controllers are finally recognized. Lastly, method (E) aligns the color tones, which further increases the detection confidence.

All these methods help the model handle the variety of clothing, poses, and lighting conditions often seen in real-world settings, highlighting the strength of the proposed approach.

Figure 9. Qualitative comparison of detection results. The columns (A) to (E) represent incremental integration of the proposed methods. Red boxes indicate incorrect detections, while green boxes represent successful detections. From left to right, we observe several red boxes turning into green boxes as additional modules are introduced, indicating more robust and generalized detection.

Figure 10 shows the qualitative results of the proposed methods on the HTC600 test dataset, as well as the results of the FPD [12] method. As shown in Figure 10, our method finds small instances (first and second rows) and effectively demonstrates its ability to detect challenging scenarios. Additionally, in the third row, our method successfully detects a Human Traffic Controller (HTC) that the FPD method fails to identify. Furthermore, in the fourth row, our method avoids false positives by not detecting a mannequin HTC, which the FPD method incorrectly identifies as an actual HTC.

Figure 10. Qualitative results on HTC600 test dataset. From left to right, each row shows the Ground Truth (GT), the method proposed by FPD [12], and Proposed (ours). For each result on the FPD and ours, we denote the number of False Positives and False Negatives.

4.4. Effect of Daylight

In this subsection, we analyze the effect of daylight on the detection performance. We build a hierarchical structure in our HTC600 training set by splitting it into two parts: one is captured under sufficient daylight (i.e., outdoor), and the other one is captured under low daylight (i.e., tunnel). Among 992 road images, 796 images were taken outdoors, and 196 images were taken in a tunnel. Examples of outdoor and tunnel scenes are given in Figure 11.

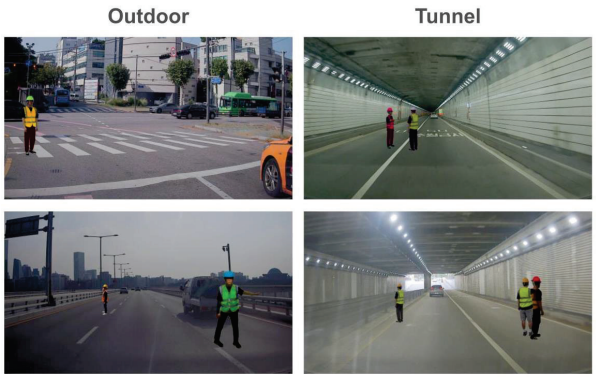


Figure 11. An example of synthesized training dataset in outdoor and tunnel scenes.

With this split, we train YOLOv7, and the results for the two different daylight conditions are given in Table 5.

As shown in Table 5, daylight had a marginal impact on performance, with the difference between outdoor and tunnel conditions being approximately 2%. This result suggests that our proposed method is robust to varying daylight conditions and is not significantly affected by changes in illumination. The minimal performance gap demonstrates the

adaptability of our approach in handling both sufficient daylight and low-light scenarios effectively, ensuring consistent detection performance across diverse environments.

Table 5. Results on two different daylight conditions.

Method	YOLOv7	
	AP_{50} (%)	AP (%)
Outdoor	56.7	31.9
Tunnel	54.3	30.8

4.5. Practical Implementations

In this subsection, we discuss the real-world applications of our proposed model in diverse traffic scenarios, such as rural and urban areas. By leveraging the diverse road background images used in our dataset generation process, the model is equipped to handle a variety of environmental conditions. Specifically, our method identifies pasteable regions within each scene and accurately synthesizes a human traffic controller, ensuring the realism and adaptability of the generated images. Examples of these synthesized images are presented in Figure 12.

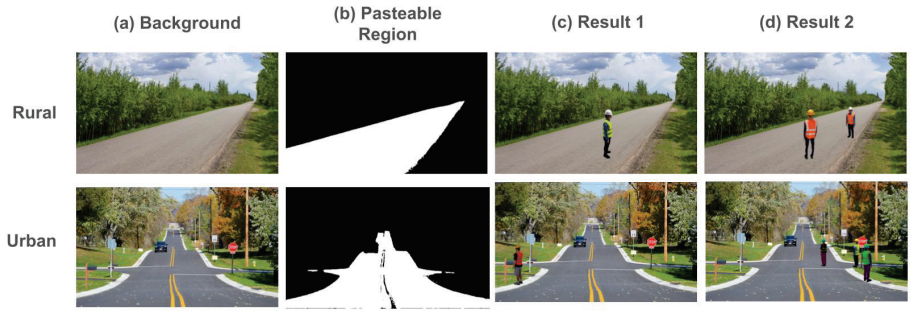


Figure 12. Examples of synthesized images across diverse traffic scenarios. Each row represents a different environment (rural on the top row and urban on the bottom row). Each column shows (**a**) the original background image, (**b**) the pasteable region identified by our method, (**c**) the first synthesized result, and (**d**) the second synthesized result.

As shown in Figure 12, our method can generate well-synthesized images across a variety of traffic scenarios, effectively placing a human traffic controller in different environments without compromising visual realism. By leveraging diverse background images, our approach ensures the adaptable and consistent incorporation of new elements into each scene. This adaptability not only broadens the applicability of the synthesized images for training intelligent transportation systems but also enhances the overall robustness of the model in handling real-world complexities.

4.6. Computational Complexity Analysis

To evaluate the capability of YOLOv7 in real-time scenarios, we measured the inference time varying numbers of Human Traffic Controllers (HTCs) using the HTC600 training dataset. The dataset includes scenarios with 1–3 HTCs. As shown in Table 6, our results confirm that the model maintains real-time performance for up to three HTCs.

Table 6. Inference time based on the number of detected Human Traffic Controllers (HTCs).

Number of Detections	Mean Inference Time (ms)	Mean NMS Time (ms)
1	5.87	0.98
2	5.91	0.98
3	5.87	0.98

We further expanded our computational complexity analysis by including both the mean detection time and its standard deviation, as summarized in Table 7. This additional information provides deeper insight into how our model can be deployed in practical real-time scenarios.

Table 7. Detection time statistics for YOLOv7 (Batch Size = 1).

Metric	Mean Detection Time (ms)	Standard Deviation (ms)
Inference Time	5.9	1.2
NMS Time	1.0	0.1
Total Detection Time	6.9	–

In addition, we analyzed the floating point operations (FLOPs) and parameter count of YOLOv7, as summarized in Table 8. This breakdown highlights the computational demands and memory footprint of the model, which are key factors in determining its suitability for real-time deployment.

Table 8. FLOPs and parameters of YOLOv7.

Model	FLOPs (G)	Number of Parameters (M)
YOLOv7	103.2	36.5

Overall, the results of these extended analyses confirm that YOLOv7 can operate in real time, making it a viable candidate for real-world applications.

5. Conclusions

In this paper, we introduced a novel approach for generating synthetic training data to enhance Human Traffic Controller (HTC) detection at road construction sites. We improved HTC detection performance by combining diffusion-based generation, color boosting, perspective adjustments, and color harmonization. Our method achieved an AP_{50} score of 73.9%, surpassing the baseline and demonstrating the effectiveness of synthetic data augmentation in diverse and challenging real-world scenarios. To validate our approach, we developed the HTC600 dataset, which includes real-world images, internet-sourced crops, and synthetic variations. This dataset has been made publicly available to support ongoing research in this domain. While our study effectively addresses key challenges related to data scarcity and appearance variability in HTC detection, it assumes that HTCs control traffic using only their bare hands. This limitation excludes scenarios where HTCs use tools, such as traffic control batons or reflective paddles, which are frequently encountered in real-world settings. To overcome this limitation, future research will focus on expanding the dataset to include tool-assisted traffic control scenarios and developing synthesis methods that enable the generation of HTCs using such tools. These advancements will capture a broader range of real-world conditions and further enhance the robustness and applicability of our detection models.

To make our system safer and avoid misuse in critical traffic conditions, we can explore additional measures in future work. One example is to combine our HTC detection with

construction site recognition in [11]. By verifying that an HTC is detected and a genuine construction zone is present, we can prevent unnecessary responses in places where no real construction is happening. This approach makes our system both more reliable and more practical in real-world situations.

By addressing these safety considerations, our study contributes to the advancement of autonomous driving technologies by tackling one of the many corner cases that must be resolved to ensure safe operation in Level 4 autonomous vehicles. Unlike Level 3 systems, which rely on user intervention in challenging scenarios, Level 4 systems operate without any human input, necessitating robust solutions for all potential corner cases. Among these, accurately detecting human traffic controllers during road construction represents a critical scenario for ensuring reliable and safe navigation in dynamic, human-involved road environments. By providing a solution to this specific corner case, our work contributes to the broader goal of enabling autonomous vehicles to navigate safely and reliably across diverse and complex real-world conditions. Furthermore, we emphasize that additional efforts should be made to compress the detector model and reduce its computational load as part of future work.

Overall, our work advances autonomous driving by improving the reliable detection of human traffic controllers, a critical component for ensuring safety in dynamic road environments. By leveraging synthetic data generation, we aim to enhance the safety and reliability of autonomous systems operating in complex and diverse settings.

Author Contributions: Conceptualization, S.B. and E.K.; data curation, S.B.; formal analysis, S.B. and E.K.; funding acquisition, E.K.; investigation, S.B. and E.K.; methodology, S.B. and E.K.; project administration, E.K.; resources, S.B.; software, S.B.; supervision, E.K.; validation, S.B.; visualization, S.B.; writing—original draft, S.B.; writing—review and editing, E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology Innovation Program (No. 20014124, Development of multipurpose mid-size bus platform technology for automated driving based on predefined route) funded By the Ministry of Trade, Industry and Energy (MOTIE), Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
2. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Virtually, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229. [CrossRef]
3. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974. [CrossRef]
4. Reutov, I. Generating of synthetic datasets using diffusion models for solving computer vision tasks in urban applications. *Procedia Comput. Sci.* **2023**, *229*, 335–344. [CrossRef]
5. Zhu, J.; Li, S.; Liu, Y.; Huang, P.; Shan, J.; Ma, H.; Yuan, J. ODGEN: Domain-specific Object Detection Data Generation with Diffusion Models. *arXiv* **2024**, arXiv:2405.15199. [CrossRef]
6. Chen, Z.; Yang, J.; Feng, Z.; Zhu, H. RailFOD23: A dataset for foreign object detection on railroad transmission lines. *Sci. Data* **2024**, *11*, 72. [CrossRef]

7. Orić, M.; Galić, V.; Novoselnik, F. Synthetic dataset generation system for vehicle detection. *Softw. Impacts* **2024**, *23*, 100735. [CrossRef]
8. Goyal, M.; Mahmoud, Q.H. A systematic review of synthetic data generation techniques using generative AI. *Electronics* **2024**, *13*, 3509. [CrossRef]
9. Lee, S.; Hyun, J.; Seong, H.; Kim, E. Unsupervised domain adaptation for semantic segmentation by content transfer. In Proceedings of the AAAI conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 8306–8315. [CrossRef]
10. Eversberg, L.; Lambrecht, J. Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization. *Sensors* **2021**, *21*, 7901. [CrossRef]
11. Seong, H.; Baik, S.; Lee, Y.; Lee, S.; Kim, E. Content swapping: A new image synthesis for construction sign detection in autonomous vehicles. *Sensors* **2022**, *22*, 3494. [CrossRef] [PubMed]
12. Lee, S.; Lee, S.; Seong, H.; Hyun, J.; Kim, E. Fallen person detection for autonomous driving. *Expert Syst. Appl.* **2023**, *213*, 119242. [CrossRef]
13. Bhunia, A.K.; Khan, S.; Cholakkal, H.; Anwer, R.M.; Laaksonen, J.; Shah, M.; Khan, F.S. Person image synthesis via denoising diffusion model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5968–5976. [CrossRef]
14. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104. [CrossRef]
15. Son, H.; Kim, C. Integrated worker detection and tracking for the safe operation of construction machinery. *Autom. Constr.* **2021**, *126*, 103670. [CrossRef]
16. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
17. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
18. Park, M.; Bak, J.; Park, S.; et al. Small and overlapping worker detection at construction sites. *Autom. Constr.* **2023**, *151*, 104856. [CrossRef]
19. Ma, L.; Jia, X.; Sun, Q.; Schiele, B.; Tuytelaars, T.; Van Gool, L. Pose guided person image generation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
20. Esser, P.; Sutter, E.; Ommer, B. A variational u-net for conditional appearance and shape generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8857–8866. [CrossRef]
21. Siarohin, A.; Sangineto, E.; Lathuiliere, S.; Sebe, N. Deformable gans for pose-based human image generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3408–3416. [CrossRef]
22. Ren, Y.; Yu, X.; Chen, J.; Li, T.H.; Li, G. Deep image spatial transformation for person image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 16–18 June 2020; pp. 7690–7699. [CrossRef]
23. Men, Y.; Mao, Y.; Jiang, Y.; Ma, W.Y.; Lian, Z. Controllable person image synthesis with attribute-decomposed gan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 16–18 June 2020; pp. 5084–5093. [CrossRef]
24. Zhang, J.; Li, K.; Lai, Y.K.; Yang, J. Pise: Person image synthesis and editing with decoupled gan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 19–25 June 2021; pp. 7982–7990. [CrossRef]
25. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv* **2023**, arXiv:2303.05499. [CrossRef]
26. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. Sam 2: Segment anything in images and videos. *arXiv* **2024**, arXiv:2408.00714. [CrossRef]
27. Fang, G.; Ma, X.; Song, M.; Mi, M.B.; Wang, X. Depgraph: Towards any structural pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 16091–16101. [CrossRef]
28. Günay, B.; Okcu, S.B.; Bilge, H.Ş. LPYOLO: Low precision YOLO for face detection on FPGA. *arXiv* **2022**, arXiv:2207.10482. [CrossRef]
29. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
30. Kong, H.; Audibert, J.Y.; Ponce, J. Vanishing point detection for road detection. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: New York, NY, USA, 2009; pp. 96–103. [CrossRef]
31. Yun, W.H.; Kim, T.; Lee, J.; Kim, J.; Kim, J. Cut-and-paste dataset generation for balancing domain gaps in object instance detection. *IEEE Access* **2021**, *9*, 14319–14329. [CrossRef]

32. Caprile, B.; Torre, V. Using vanishing points for camera calibration. *Int. J. Comput. Vis.* **1990**, *4*, 127–139. [CrossRef]
33. Diederik, P.K. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
34. Khanam, R.; Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

An Efficient and Low-Complexity Transformer-Based Deep Learning Framework for High-Dynamic-Range Image Reconstruction

Josue Lopez-Cabrejos *, Thuanne Paixão, Ana Beatriz Alvarez and Diodomiro Baldomero Luque

PAVIC Laboratory, University of Acre (UFAC), Rio Branco 69915-900, Brazil; thuannepaixao@gmail.com (T.P.); ana.alvarez@ufac.br (A.B.A.); baldomero.luque@ufac.br (D.B.L.)
* Correspondence: josue.cabrejos@sou.ufac.br

Abstract: High-dynamic-range (HDR) image reconstruction involves creating an HDR image from multiple low-dynamic-range images as input, providing a computational solution to enhance image quality. This task presents several challenges, such as frame misalignment, overexposure, and motion, which are addressed using deep learning algorithms. In this context, various architectures with different approaches exist, such as convolutional neural networks, diffusion networks, generative adversarial networks, and Transformer-based architectures, with the latter offering the best quality but at a high computational cost. This paper proposes an HDR reconstruction architecture using a Transformer-based approach to achieve results competitive with the state of the art while reducing computational cost. The number of self-attention blocks was reduced for feature refinement. To prevent quality degradation, a Convolutional Block Attention Module was added, enhancing image features by using the central frame as a reference. The proposed architecture was evaluated on two datasets, achieving the best results on Tel's dataset in terms of quality metrics. The computational cost indicated that the architecture was significantly more efficient than other Transformer-based approaches for reconstruction. The results of this research suggest that low-complexity Transformer-based architectures have great potential, with applications extending beyond HDR reconstruction to other domains.

Keywords: high-dynamic-range imaging; image reconstruction; channel attention; spatial attention; Transformers

1. Introduction

Capturing information faithfully from a scene using conventional cameras is a complex task. Unlike human eyes, a camera compresses the high dynamic range (HDR) of the scene to capture and display it in a low-dynamic-range (LDR) image [1]. This not only limits the luminance range in the captured image but also causes information outside of this range not to be captured in the final image [2]. To address this, specialized hardware has been developed to directly produce an HDR image; however, these devices tend to be very expensive for mass adoption [3,4].

Initially, methodologies based on conventional algorithms were developed to convert LDR images into HDR images, using algorithms to address issues related to object movement in the scene [5–8]. Other approaches focus on removing defects caused by camera movement [9–12], and some algorithms employ a patch-based methodology to align the input images [13–15]. In recent years, algorithms to convert a single LDR image into an

HDR image have been developed. Some methodologies rely on inverse tone mapping [16], while other architectures use deep learning algorithms based on convolutional neural networks (CNNs) [17]. However, results using only a single LDR input image exhibit incomplete information, which can be visually apparent.

With this challenge in mind, a common methodology to maintain the fidelity of the original scene is to capture a series of images at different exposures and subject them to a fusion process, thereby reconstructing the scene with the highest possible fidelity and obtaining an HDR image [13,18]. This approach has particular characteristics. The ideal case is when three images with different exposures are captured at the same moment in time, in which case the only difference between the images is the exposure value. In this scenario, the images would not suffer from misalignment, and consequently, no artifacts would be generated in the HDR reconstruction. In typical scenarios, the images are captured consecutively, making them prone to various types of degradation, such as misalignment, motion, noise, and overexposure, all of which must be corrected during the HDR reconstruction process [19,20]. An approach currently used to perform this reconstruction is deep learning. Some methods employ alignment mechanisms between frames before image fusion [21,22], while others use temporal or spatial attention mechanisms to highlight regions within the frames and then perform the necessary fusion to reconstruct the HDR image [23–25]. In both contexts, an image restoration architecture is used to perform the frame fusion and handle the degradation. Typically, conventional methods such as CNN-based architectures are used for this purpose [26–28], while more recent approaches employ diffusion-based architectures [29], Generative Adversarial Networks (GANs) [30], and Transformers [31,32].

Regarding Transformers, this architecture was initially used for natural language processing (NLP), being a complete architecture with an encoder–decoder approach, utilizing self-attention to capture long-range dependencies within the processed text [33]. Its domain of use was quickly expanded to include image-related tasks, where only the encoder portion of NLP Transformers was retained. The functioning of this encoder component begins by dividing the image into small patches, which are treated as linear vectors and encoded using position embeddings to retain information about the positions of these patches relative to the original image. These patches are then fed into a series of encoder blocks composed of layer normalization (LN), self-attention, LN, and a multilayer perceptron (MLP). After this process, the patches are reassembled according to their original positions to maintain the original resolution [34].

Transformer-based architectures incur a high computational cost [35] due to the recursive application of self-attention across the entire image [34], resulting in quadratic complexity. This complexity is offset by their powerful encoding capability, which improves state-of-the-art performance in HDR reconstruction tasks. However, as the architecture becomes more complex, the computational cost increases significantly. Consequently, current works in HDR reconstruction focus on achieving the best results in terms of quality metrics of the reconstructed image compared to the original, such as the Peak Signal-to-Noise Ratio (PSNR) or the Structural Similarity Index Measure (SSIM) [36], often overlooking improvements in memory consumption, decreases in processing time, or reductions in the number of parameters [37]. Among Transformer-based HDR reconstruction approaches, SCTNet, proposed by Tel. et al. [32], stands out, as it employs self-attention and cross-attention mechanisms to enhance the features extracted from the input image, achieving the best quality results in the state of the art. When comparing the computational cost of SCTNet with those of previous Transformer-based proposals, a slight improvement is observed because other architectures continue to present high inference times. However, when comparing inference time between SCTNet and other approaches, such as GANs, the SCTNet

approach demonstrates inferior results, highlighting the fact that this architecture was not developed with the goal of reducing computational cost.

To maintain good results in terms of quality metrics and improve computational cost, a new HDR reconstruction architecture based on Transformers is proposed. This architecture is based on the SCTNet methodology, where the typical number of self-attention blocks in Transformers is reduced from twenty-four to six self-attention blocks, while two refinement blocks called Convolutional Block Attention Modules (CBAM) blocks [38] are added. The CBAM blocks consist of channel-level attention followed by spatial-level attention, referencing the central exposure image and performing cross-attention with an image of different exposure. The efficiency of this architecture in terms of reconstruction quality is demonstrated by the good quantitative results obtained with the PSNR, SSIM, and HDR-VDP-2 metrics, as well as the achieved qualitative results. The experiments were conducted using the datasets from Kalantari et al. [39] and Tel et al. [32]. Additionally, the results regarding computational cost, calculated from the numbers of parameters and multiply–accumulate units; the model size; and the inference time, are important to highlight, as they demonstrate that the proposed architecture is faster than other Transformer-based models. Thus, the main contributions of this paper can be summarized as follows:

- We propose a new high-quality HDR reconstruction architecture based on Transformers that has a lower computational cost than current state-of-the-art methodologies.
- We introduce the use of CBAM blocks to enhance input features in the self-attention mechanisms of the Transformer architecture for HDR reconstruction.
- Exhaustive experiments are conducted to demonstrate the effectiveness of the proposed HDR reconstruction. Two main aspects are explored: the computational cost required and the quality of the HDR output image, considering both quantitative and qualitative comparisons.

2. Related Works

A widely used approach is to reconstruct an HDR image from a single LDR image, which offers advantages such as the absence of misalignment and blur. However, this method introduces other types of defects known as artifacts [40]. Current methodologies attempt to address these artifacts using deep learning-based architectures, such as HDRUNet [17] and DCDR-UNet [41], which employ image restoration architectures based on multi-scale convolutions and Deformable Convolution Residual Blocks, respectively. Datasets developed for this type of reconstruction typically feature three input frames, as proposed by NTIRE [37] in 2021, but only one of these frames is used for training and testing the architectures.

Currently, the most widely used approach for HDR reconstruction involves creating an HDR image from LDR images with different exposures, known as multi-frame HDR reconstruction. Tursun et al. [42] identify ghosting artifacts as the main challenges in HDR reconstruction, including light-source motion, object movement, object deformation, and object occlusion. These challenges have led to the development of various feature extraction architectures and blocks based on CNN architectures [26,39], diffusion-based architectures [29], and innovative methods using Transformers [31,32]. A common theme among these methods is the extraction of features from multiple images and their fusion using image restoration architectures, achieving high-quality HDR reconstruction. Feature extraction is typically performed using alignment mechanisms [43] or attention blocks [44], where the central exposure image serves as a reference for applying the desired alignment or attention. The dataset commonly used in this type of reconstruction is the one proposed by Kalantari et al. [39] in 2017, though new proposals inspired by this dataset, such as the one by Tel et al. [32] in 2023, also exist.

Transformer-based models are widely used for HDR tasks due to their strong image reconstruction capabilities. The SwinIR architecture [45] was used by Liu et al. [31] in HDR-Transformer, adding a frame alignment mechanism before feature extraction. On the other hand, Kim et al. [46] used Transformers in a UNet form for feature alignment and refinement. Zou et al. [47] proposed HDT-HDR, which uses a dual Transformer mechanism that, in addition to performing self-attention on the input features, also performs refinement through convolutions and adds them at the end of the process, enriching the refinement. Chen et al. [48] presented a pyramid-shaped Transformer architecture called HFT-HDR for feature fusion, leaving the reconstruction task to a series of convolutional blocks. PASTA is an architecture proposed by Liu et al. [49] that performs multi-scale feature refinement using SwinIR-based Transformer layers in conjunction with channel-level attention. Yan et al. [50] proposed HyHDRNet, which consists of two networks: one focuses on removing ghost artifacts from features, and the other is a SwinIR-based network that enhances these features. Chi et al. [51] proposed SV-HDR, a novel technique using a Transformer architecture for denoising in a diffusion-based reconstruction approach, which outperforms other denoisers. Lastly, Yan et al. [52] proposed SSHDR, a semi-supervised reconstruction approach based on SwinIR using datasets with and without ground truth.

A methodology that has recently achieved excellent performance using only Transformer mechanisms is SCTNet, proposed by Tel et al. [32]. It adopts the SwinIR reconstruction architecture [45] and adds cross-attention mechanisms. The authors conducted comparative experiments using the Kalantari and Tel datasets, obtaining the best performance in HDR image reconstruction. The results were compared with proposals from Hu et al. [14] and Sen et al. [13], which are based on patch-level image fusion and use only CPUs for processing. For comparison purposes, the authors also used CNN-based architectures, including DHDRNet [39], which incorporates an alignment process for HDR reconstruction, DeepHDR [53], which aligns features using the medium-exposure image as a reference, and NHDRRNet [54], which uses non-local attention mechanisms instead of alignment to achieve the same purpose. Other architectures such as AHDRNet [55] use attention blocks for feature alignment and dense residual dilation blocks to reconstruct HDR images, while CEN-HDR [44] utilizes spatial attention to improve the extracted features with convolution mechanisms for further refinement.

Inspired by the generative approach, HDR-GAN [30] was proposed as the first architecture based on GANs for fusing multiple-exposure LDR images for HDR reconstruction. Evaluations using the Kalantari dataset demonstrated that this approach outperformed architectures such as Sen's, Hu's, and Kalantari's architectures as well as DeepHDR, AHDRNet, and NHDRRNet under the same evaluation conditions. On the other hand, DiffHDR [29] introduced a diffusion-based approach for HDR reconstruction, treating the HDR image as a conditional generative modeling task. Results on the Kalantari dataset showed good generalization for real-world images and competitive performance with state-of-the-art methodologies, including those of Hu and Kalantari as well as DeepHDR, AHDRNet, NHDRRNet, HDR-GAN, ADNet [27], APNT [56], ST-HDR [57], and HDR-Transformer [31]. In another approach, HDR-Transformer [31] replaced traditional convolutional feature encoding processes with Transformer-based attention mechanisms. The authors showed its superior performance compared to methods such as Sen's, Hu's, and Kalantari's methods as well as DeepHDR, AHDRNet, NHDRRNet, HDR-GAN, and SwinIR [45]. They also compared inference time and model parameters, highlighting the superiority of their approach in balancing performance and efficiency.

The main focus of the HDR-GAN, DiffHDR, HDR-Transformer, and SCTNet architectures is on improving reconstruction quality. However, they do not prioritize the reduction of computational cost. As a result, Transformer-based approaches show a high associated

computational cost, particularly during inference time when reconstructing an HDR image. In contrast to the state of the art, the architecture proposed in this paper primarily focuses on reducing computational cost without compromising the final HDR reconstruction quality. This approach is based on the Transformer mechanisms presented in SCTNet, where the number of original self-attention and cross-attention blocks is reduced and complemented with convolutional attention blocks to maintain reconstruction quality.

3. Proposed Method

This section is organized to address four key aspects of the proposed architecture. First, it describes the overall functioning of the architecture, explaining how the input image is processed and outlining the stages and blocks that make it up. Then, it provides a detailed explanation of the processing carried out by each block in the feature extraction stage, describing the CBAM block used in this stage and the concatenation of features after the convolution performed on each input frame. Next, this section discusses the feature refinement stage, where a Transformer-based architecture is used. The self-attention in the Global Self-Attention Block (G-SAB) and cross-attention in the Spatial Cross-Attention Block (S-CAB) applied to the input are explained, along with the additional feature fusion known as the skip connection. Finally, the loss function used during the training of the architecture is detailed, explaining the equations that make up this function.

3.1. Overall Architecture

The input LDR images are three-channel images (red, green, and blue). However, to enhance the efficiency of HDR reconstruction, the HDR equivalent for each frame is computed by applying gamma encoding, as described in Equation (1).

$$H_i = \frac{L_i^{\gamma}}{t_i} \tag{1}$$

where $L_i \in \mathbb{R}^{3\times H\times W}$ corresponds to each input frame $i \in \{1,2,3\}$; $\gamma = 2.2$ is a typical value for gamma correction; t_i is the exposure time of the frame used; and $H_i \in \mathbb{R}^{3\times H\times W}$ is the result of applying gamma correction, which is still a three-channel image. Therefore, our input, denoted as $I_i \in \mathbb{R}^{6\times H\times W}$, becomes a six-channel image, resulting from the concatenation of L_i with H_i, as shown in Equation (2).

$$I_i = [L_i\, H_i] \tag{2}$$

The proposed architecture is shown in Figure 1. The reconstruction process begins with the feature extraction stage, where three LDR input frames with different exposures are used. The exposure values may vary depending on the dataset, but the frame with the median exposure is always used as the reference for the spatial position of the objects. Next, both channel-level and spatial-level attention are applied using the CBAM block. This attention process highlights regions of the image, meaning it gives more importance to the areas that the model considers relevant in the high- or low-exposure frames with respect to the central frame.

Subsequently, a convolution block is applied to each frame to extract features from each image. These features are concatenated to be refined by a Transformer architecture. The next stage is feature refinement, which begins with an MLP block that enables the fusion of the non-linear features present in the input. This results in optimized features to be refined with the G-SAB block, which performs self-attention on these features, followed by the S-CAB block. The S-CAB block splits the features into groups of three to apply cross-attention and then concatenates them again at the end of this process.

The output of this refinement is filtered with a convolution and added to the initial features of the central frame in a process called skip connection. Finally, all the obtained features are decoded with a final convolution to reconstruct the corresponding HDR image.

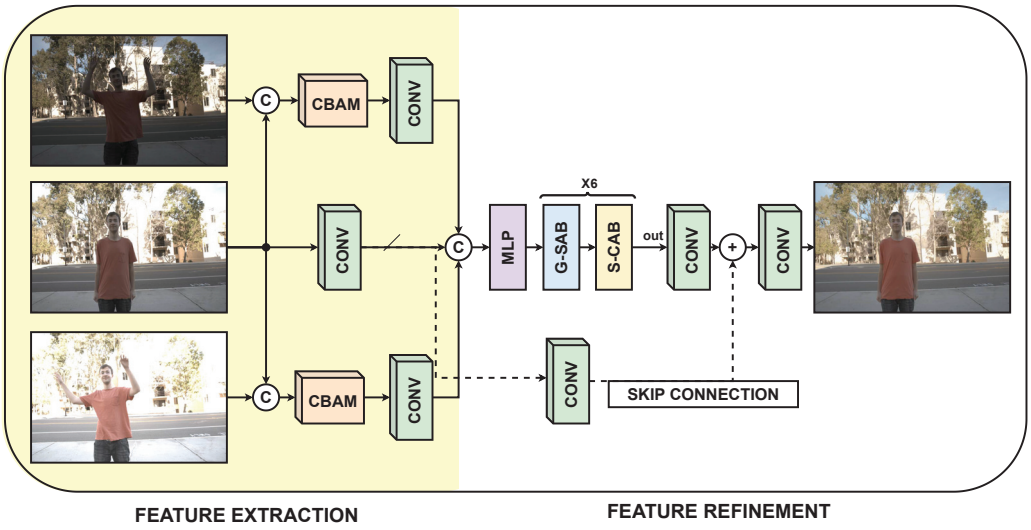


Figure 1. Proposed architecture for HDR reconstruction consisting of two stages; the first stage is feature extraction, and the second stage is feature refinement. The proposed architecture contains two CBAM blocks, representing consecutive channel-level attention and spatial-level attention; convolution blocks; and a Transformer architecture adopted from Tel et al. [32].

3.2. Feature Extraction

3.2.1. CBAM

The frames I_1 and I_3 correspond to high- and low-exposure frames, respectively. The CBAM block takes as input one of these frames along with the reference frame I_2. Let the concatenation of both images be denoted as $F \in \mathbb{R}^{12\times H\times W}$. In general, the attention process in CBAM can be described by Equation (3).

$$\begin{aligned} F^{'} &= M_C(F) \otimes F \\ F^{''} &= M_S(F^{'}) \otimes F^{'} \end{aligned} \tag{3}$$

where M_C represents the channel-level attention, M_S represents the spatial-level attention, $F^{'}$ is the output of the channel-level attention, and $F^{''}$ is the output of the spatial-level attention. The symbol $\otimes$ denotes an element-wise operation. Figure 2 illustrates the process of the input image through the CBAM, which consists of the Channel Attention Module (CAM) and the Spatial Attention Module (SAM).

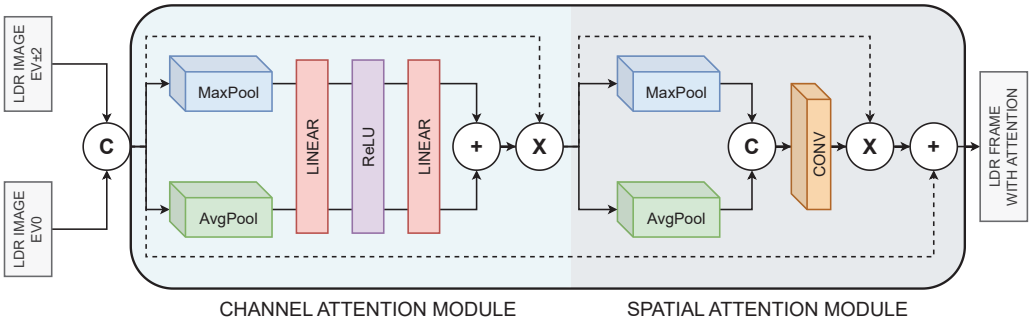


Figure 2. Composition of the CBAM block, consisting of two modules. The first one receives two concatenated images to apply channel-level attention, then applies spatial-level attention. The output is an image with both types of attention applied.

The CAM module generates an attention map that considers the channel-level relationships of the input features F. The attention map highlights the most important regions in the image. To achieve this, one channel is computed using Max Pooling and another using Average Pooling. Each channel is processed by an MLP with a ReLU activation function, where the MLP is shared by the two channels, meaning the same weights are used for channel-level attention. After this process, the two channels are summed to obtain the final attention map M_C, as described by Equation (4), where *AvgPool* refers to Average Pooling, *MaxPool* refers to Max Pooling, and σ represents the sigmoid function. This map is then multiplied by the input, generating a channel-attended image.

$$M_C(F) = \sigma(\text{MLP}(AvgPool(F)) + \text{MLP}(MaxPool(F))) \tag{4}$$

For the SAM module, where F' is an image with channel-level attention, the next step is to apply spatial-level attention. The first step is to calculate the Max Pooling and Average Pooling channels, which are then concatenated to generate a spatial attention map M_S using a final convolution, as described in Equation (5). The map M_S is then applied to the input image to obtain the corresponding spatial attention. Finally, this result is added to the initial input image, yielding an output image with both attentions applied.

$$M_S(F') = \sigma(\text{CONV}[AvgPool(F')\, MaxPool(F')]) \tag{5}$$

3.2.2. Feature Concatenation

Herein, F_1'' represents the low-exposure image and F_3'' represents the high-exposure image after the CBAM block, while I_2 represents the medium-exposure image. The features of each of these images are extracted using a convolutional block and then concatenated to be used as inputs for the next stage. This process is mathematically described by Equation (6).

$$z = [\text{CONV}(F_1'')\, \text{CONV}(I_2)\, \text{CONV}(F_3'')] \tag{6}$$

3.3. Feature Refinement

3.3.1. G-SAB

To distinguish the spatial relationship between different features, a multi-head self-attention technique based on windows, as proposed by Tel et al. [32], was used. To achieve this, the image is divided into small patches of 8×8 pixels, which are tokenized to feed into the G-SAB block, as shown in Figure 3. Within this block, there are four main components: initially, the input patches are normalized by Layer Normalization (LN), followed by the calculation of patch-level self-attention in the W-MSA. The result is then normalized again with another LN, and non-linear relationships are found using an MLP. This makes it possible to assign importance to specific regions of the initial features in relation to the entire feature set. This modeling can be described by considering a patch sequence z_{j-1} as input and z_j as output, as mathematically expressed in Equation (7).

$$\begin{aligned} \hat{z}_j &= \text{W-MSA}(\text{LN}(z_{j-1})) + z_{j-1} \\ z_j &= \text{MLP}(\text{LN}(\hat{z}_j)) + \hat{z}_j \end{aligned} \tag{7}$$

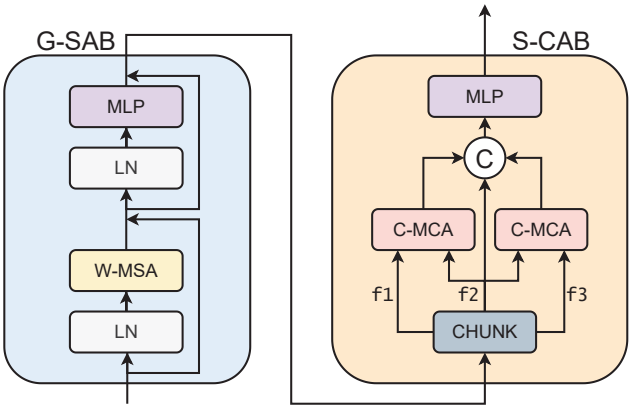


Figure 3. Two consecutive self-attention blocks. G-SAB applies conventional self-attention for the input features, while S-CAB applies cross-attention, using features of the central frame as the query and features of the other frames as the key and value.

3.3.2. S-CAB

To perform cross-attention given an input feature z_j, the feature is divided into three subgroups with an equal number of channels: f_1, f_2, and f_3, corresponding to the features of the low-, medium-, and high-exposure frames, respectively, as described in Equation (8).

$$[f_1 \ f_2 \ f_3] = z_j \tag{8}$$

Next, channel multi-head cross-attention (C-MCA) is applied between these features, using f_2 as the query q_2; f_1 generates the key and value for k_1, v_1, and f_3 generates the key and value for k_3, v_3, respectively. Therefore, the output of the S-CAB can be written as shown in Equation (9), where f'_{12} and f'_{32} are the outputs after each C-MCA.

$$\begin{aligned} f'_{12} &= \text{C-MCA}(q_2, k_1, v_1) = softmax(\frac{q_2 \times k_1^T}{\sqrt{d_k}})v_1 \\ f'_{32} &= \text{C-MCA}(q_2, k_3, v_3) = softmax(\frac{q_2 \times k_3^T}{\sqrt{d_k}})v_3 \end{aligned} \tag{9}$$

where d_k is the scaling factor.

3.3.3. Skip Connection

Considering *out* as the output of the Transformer network, there is a final convolution before the aggregation of the central features F'_2 from the medium-exposure image, as given by Equation (10), where $\hat{H}$ represents the reconstructed HDR image.

$$\hat{H} = \text{CONV}(\text{CONV}(F'_2) + \text{CONV}(out)) \tag{10}$$

3.4. Loss Function

Typically, HDR images are tone-mapped for visualization purposes, which is why it is common to calculate the loss function in the tone-mapped domain. For this, the commonly used μ-law mapping is applied, as described by Equation (11).

$$T(H) = \frac{log(1 + \mu H)}{log(1 + \mu)}, \ \ \mu = 5000 \tag{11}$$

where $T(H)$ represents the tone-mapped image, H is the input HDR image, and μ is an intensity parameter. The loss function consists of two parts; the first part is the classic $\mathcal{L}_1$

loss function, which calculates the mean absolute error (MAE), as shown in Equation (12), where $\hat{H}$ is the HDR image reconstructed by the network and H is the real HDR image.

$$\mathcal{L}_1 = \|T(H) - T(\hat{H})\|_1 \tag{12}$$

Additionally, another loss function $\mathcal{L}_p$, called the perceptual loss, is calculated. This loss function is widely used in inpainting [58] for improving visual quality. It is computed using a pre-trained VGG-16 network, where the MAEs of the feature maps Φ calculated by VGG-16 are summed. This function is described by Equation (13).

$$\mathcal{L}_p = \|\Phi(T(H)) - \Phi(T(\hat{H}))\|_1 \tag{13}$$

Thus, the final loss function $\mathcal{L}$, as characterized in Equation (14), is composed of the two previously mentioned loss functions, resulting in a robust final function that considers errors not only at the pixel level but also at the feature level. A parameter α, set to 0.01, is used to adjust the value of the perceptual loss.

$$\mathcal{L} = \mathcal{L}_1 + \alpha \cdot \mathcal{L}_p \tag{14}$$

4. Experiments and Results

4.1. Implementation Details

4.1.1. Datasets

The training was carried out using the datasets from Kalantari [39] and Tel [32]. Kalantari's dataset was selected because it is the most commonly used for HDR reconstruction tasks, and Tel's dataset was selected to complement Kalantari's. To overcome the challenge of limited training images, the image was divided into small patches of 128 × 128 pixels with a stride of 64 pixels. Data augmentation included rotation and image reflection. Additionally, separate training was performed for each dataset, resulting in two different models for comparison with the state of the art.

Tel: The dataset developed by Tel et al. [32] consists of 108 training scenes and 36 test scenes. This dataset was captured using a Nikon D700 (Nikon Corporation, Tokyo, Japan) camera, with up to nine different exposures taken to obtain the reference HDR image, and a tripod was used to ensure no movement during the capture. From the nine exposures, three images with different exposures are provided for each scene. As for the LDR images, they were captured by introducing small movements to simulate natural captures. The final dataset has a resolution of 1500 × 1000 pixels

Kalantari: In 2017, Kalantari et al. [39] developed a dataset that consists of 74 training scenes and 15 test scenes. The dataset was captured using a high-quality Canon EOS-5D Mark III (Canon Inc., Tokyo, Japan) camera. Initially, more than 100 scenes were captured in RAW format with a resolution of 5760 × 3840 pixels; they were later resized to a resolution of 1500 × 1000 pixels. During the capture process, small intentional movements between frames were introduced to simulate misalignment. This dataset utilizes three input images with different exposures.

Based on Tel et al. [32], images from the Kalantari test set exhibit a high proportion of overexposed pixels. This introduces challenges not directly related to HDR reconstruction, arising from prolonged exposure times in the reference image, minimal time differences between the exposures of the LDR images, and the limitation that three input images may be inadequate for this dataset. In contrast, Tel's test set does not present these particular issues.

4.1.2. Experimental Setup

The proposed network was implemented using PyTorch 2.4.0 with Python 3.11. The ADAM optimizer was used with a fixed learning rate of 2×10^{-4}, β_1 set to 0.9, β_2 set to 0.999, and ϵ set to 10^{-8}. The network training was carried out in the PAVIC laboratory's data center, utilizing two Nvidia HGX A100 GPUs (Nvidia Corporation, Santa Clara, CA, USA) with 40 GB of memory, a batch size of 32, and 150 training epochs. The training took approximately three days.

4.1.3. Evaluation Metrics

The performance evaluation was carried out by comparing the quality metrics PSNR and SSIM, both in the linear domain and in the logarithmic domain, denoted by l-PSNR, μ-PSNR, l-SSIM, and μ-SSIM. The notation l represents the calculation in the linear domain, i.e., the reconstructed HDR image compared with the real HDR reference, while μ represents the calculation in the mapped domain, using Equation (11) to map the reconstructed HDR image and the HDR reference image. HDR-VDP-2, a metric specifically developed to quantify quality in HDR images, was also used.

Peak Signal-to-Noise Ratio: PSNR is a metric proposed by Huynh et al. [59] that quantitatively determines the quality of an image by assessing the level of noise between two signals, measured in decibels (dB). A value of 0 dB indicates that the signals are completely different, while a higher value indicates less distortion between the two signals. Mathematically, PSNR is given by Equation (15).

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{\text{MSE}}\right) \tag{15}$$

The MSE is described by Equation (16), which calculates the squared difference between the values of two signals. In the case of images, it measures the squared difference of the pixel values, where y is the original image or ground truth and $\hat{y}$ is the reconstructed image.

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

A high PSNR indicates that the reconstruction is of high quality. However, there are cases where visually similar images may have a low PSNR, as simple pixel errors might not adequately capture the structural similarity.

Structural Similarity Index Measure: The SSIM metric, proposed by Sara et al. [60], measures the structural similarity between an image x and another image y. Mathematically defined by Equation (17), it calculates the average pixel values for both images, denoted as μ_x and μ_y, respectively. The variances σ_x^2 and σ_y^2, as well as the covariance σ_{xy} between the two images, are also computed. The constants c_1 and c_2 are defined as $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, with $k_1 = 0.01$, $k_2 = 0.03$, and $L = 2^{\text{bits per pixel}} - 1$.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{17}$$

HDR-VDP-2: Using deep learning-based algorithms, Mantiuk et al. [61] introduced a model capable of discriminating quality between two HDR images, taking into account real-world conditions such as luminance, dynamic range, and color levels. The HDR-VDP-2 model was trained using the LIVE and TID2008 datasets to assess image quality, acknowledging that the comparison of HDR images involves both pixel value measurements and the properties of the display panel. The authors assume the characteristics of a standard LCD screen for this purpose, demonstrating through rigorous experimentation

that HDR-VDP-2 was calibrated and reliably predicts HDR image quality by quantifying visual differences into a numerical value.

4.2. Evaluation and Comparison

A comparative evaluation was conducted using the Tel [32] and Kalantari [39] datasets. The analysis considered prominent state-of-the-art architectures based on GANs, diffusion, and Transformers, including HDR-GAN [30], Diff-HDR [29], HDR-Transformer [31], and SCTNet [32].

4.2.1. Quantitative Analysis

For the quantitative evaluation using the Tel dataset, it was necessary to train the Diff-HDR and HDR-GAN architectures to compute the metrics. The metrics for the HDR-Transformer and SCTNet architectures were obtained from SCTNet [32]. The results are presented in Table 1.

Table 1. Comparative table of image quality metrics for architectures using the Tel dataset.

Method	μ-PSNR ↑	l-PSNR ↑	μ-SSIM ↑	l-SSIM ↑	HDR-VDP-2 ↑
HDR-GAN [30]	40.32	44.35	0.9833	0.9913	67.89
Diff-HDR [29]	42.33	46.79	0.9849	0.9948	68.91
HDR-Transformer [31]	42.39	46.35	0.9844	0.9948	69.23
SCTNet [32]	<u>42.55</u>	**47.51**	<u>0.9850</u>	<u>0.9952</u>	**70.66**
Proposed method	**43.16**	<u>47.46</u>	**0.9876**	**0.9956**	<u>69.76</u>

Note: The best result is shown in **bold**, and the second best is <u>underlined</u>. ↑ indicates that higher values are better.

In terms of μ-PSNR, μ-SSIM, and l-SSIM, the proposed architecture achieved the best results, followed by SCTNet. The results for l-PSNR and HDR-VDP-2 place the proposed method in second place, as it was outperformed by SCTNet. On the other hand, HDR-Transformer ranks as the third best in performance. Thus, the results obtained with this dataset demonstrate that the proposed architecture delivers the best performance with minimal distortion and better structural similarity between the reconstructed image and its reference.

For the evaluation using the Kalantari dataset, the values of the metrics obtained by Tel et al. [32] and Diff-HDR [29] are considered. The results are shown in Table 2.

Table 2. Comparative table of image quality metrics for architectures using the Kalantari dataset.

Method	μ-PSNR ↑	l-PSNR ↑	μ-SSIM ↑	l-SSIM ↑	HDR-VDP-2 ↑
HDR-GAN [30]	43.92	41.57	0.9905	0.9865	65.45
Diff-HDR [29]	44.11	41.73	0.9911	0.9885	65.52
HDR-Transformer [31]	<u>44.32</u>	<u>42.18</u>	0.9916	0.9884	66.03
SCTNet [32]	**44.49**	**42.29**	**0.9924**	**0.9887**	<u>66.65</u>
Proposed method	44.10	42.07	<u>0.9917</u>	<u>0.9886</u>	**67.79**

Note: The best result is shown in **bold**, and the second best is <u>underlined</u>. ↑ indicates that higher values are better.

The results show the superiority of SCTNet in almost all metrics. When considering the image quality metric for reconstructed images under real-world conditions, HDR-VDP-2, the proposed methodology outperforms all other architectures.

4.2.2. Qualitative Analysis

A detailed visual comparison was performed on images from the Tel [32] and Kalantari [39] datasets, specifically using scenes 29 and 34 from the Tel dataset and scenes 9 and 7 from the Kalantari dataset. Each of the analyzed scenes will be presented with images

arranged in three columns: the input LDR images, the reconstructed image, and the area for detailed analysis.

Scene 29 from the Tel dataset is presented in Figure 4a, where the three input LDR images, the HDR image reconstructed by the proposed method, and the area for detailed analysis are shown. The selected area, highlighted by a red box, exhibits overexposure in the input LDR images, making it a challenging scene to reconstruct.

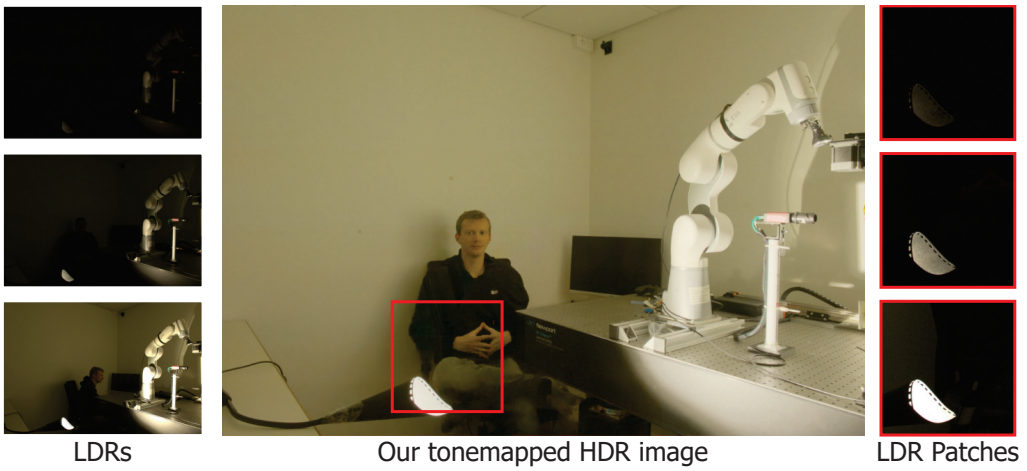
Figure 4b shows the areas of interest cropped from the images reconstructed by the analyzed architectures. Due to the complexity of the scene, the presence of ghosting artifacts becomes evident in all the reconstructions, with all architectures facing difficulties in performing the reconstruction. Additionally, HDR-GAN and SCTNet exhibit a blur effect observed in the vertical structure, indicated by an arrow. Diff-HDR and HDR-Transformer show a darker tone than the reference. In contrast, the reconstruction by the proposed method produces the closest color to the reference while also reducing the intensity of the ghosting artifact.

Figure 4. Qualitative comparison of the reconstruction of scene 29 from the Tel dataset [32]. (**a**) Input images with different exposures and the HDR image reconstructed by the proposed method, with the analyzed region highlighted by a red box. (**b**) HDR image reconstructions by HDR-GAN [30], Diff-HDR [29], HDR-Transformer [31], SCTNet [32], and the proposed methodology, followed by the reference image. The red arrows indicate the main differences in the images.

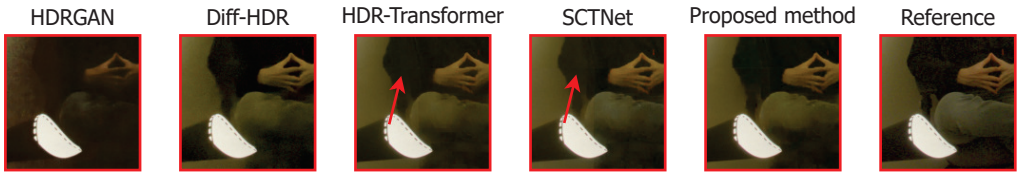
Similarly, Figure 5a illustrates scene 34 from the Tel dataset, where the selected area for analysis is highlighted by a red box. This area presents a challenge for reconstruction due to the lack of information in the input LDR images caused by underexposure, as observed in the LDR patches.

Figure 5b presents the analysis areas extracted from the images reconstructed by each architecture. The reconstruction performed by HDR-GAN shows color discrepancies compared to the reference. The result achieved by Diff-HDR exhibits inconsistencies in its reconstruction, presenting a significant amount of noise. HDR-Transformer and SCTNet show similar characteristics regarding the blur effect, which is more intensely observed in the region indicated by the red arrow. On the other hand, the results obtained by the

proposed method demonstrate a reconstruction with more appropriate color contrast than the reference.



(**a**)



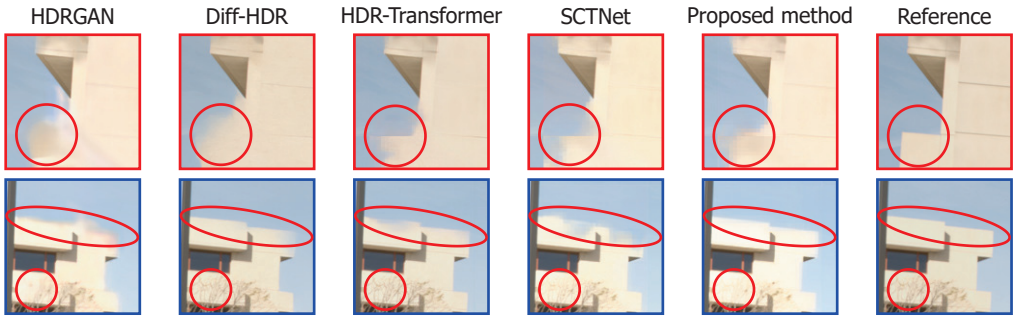
(**b**)

Figure 5. Qualitative comparison of the reconstruction of scene 34 from the Tel dataset [32]. (**a**) Input images with different exposures and the HDR image reconstructed by the proposed method, with the analyzed region highlighted by a red box. (**b**) HDR image reconstructions by HDR-GAN [30], Diff-HDR [29], HDR-Transformer [31], SCTNet [32], and the proposed methodology, followed by the reference image. The red arrows indicate the main differences in the images.

Figure 6a illustrates scene 9 from the Kalantari dataset, a figure with overexposure of objects that poses a challenge for HDR reconstruction. Two regions of this scene were analyzed, with the areas marked by the red box and the blue box highlighting regions where the information in the different LDR frames is scarce and where there is high contrast in the edges and walls of the building, showing a complex scenario for reconstruction.

The reconstruction behaviors of the different analyzed architectures for the demarcated areas are presented in Figure 6b. In the HDR reconstruction for the red-marked region containing building walls with a sky background, HDR-GAN and Diff-HDR architectures show more difficulty in reconstructing details of the analyzed physical structure, as highlighted within the red circle. In contrast, HDR-Transformer, SCTNet, and the proposed method achieves better reconstruction of the building walls in the same region. In the blue-marked region, the red details show that the roof of the building and the tree branches in the reconstructed images are inadequately captured in the HDR-GAN and Diff-HDR reconstructions compared to the reference image. However, HDR-Transformer, SCTNet, and the proposed method presents better reconstructions of these details. These results highlight the efficiency of Transformer-based architectures for HDR reconstruction compared to others.

(**a**)



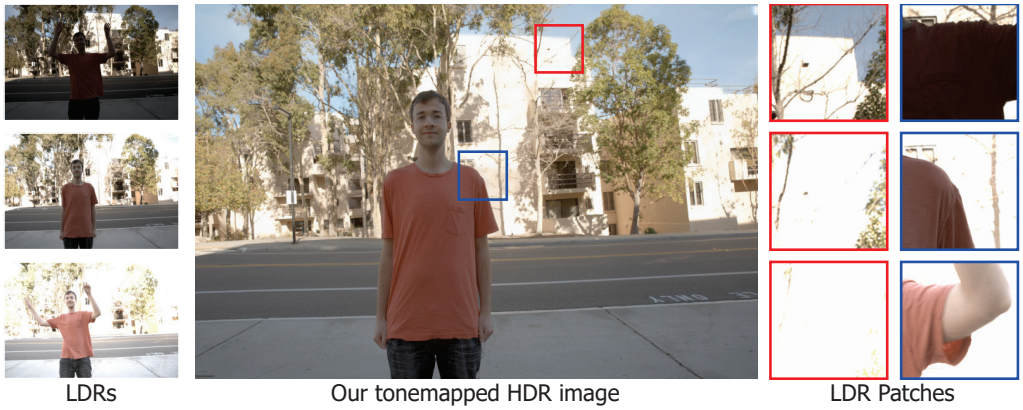
(**b**)

Figure 6. Qualitative comparison of the reconstruction of scene 9 from the Kalantari dataset [39]. (**a**) Input images with different exposures and the reference HDR image, with red and blue boxes highlighting the analyzed regions. (**b**) HDR image reconstructions by HDR-GAN [30], Diff-HDR [29], HDR-Transformer [31], SCTNet [32], and the proposed methodology, followed by the reference image. The red circles highlight the main challenges in the reconstructed images.

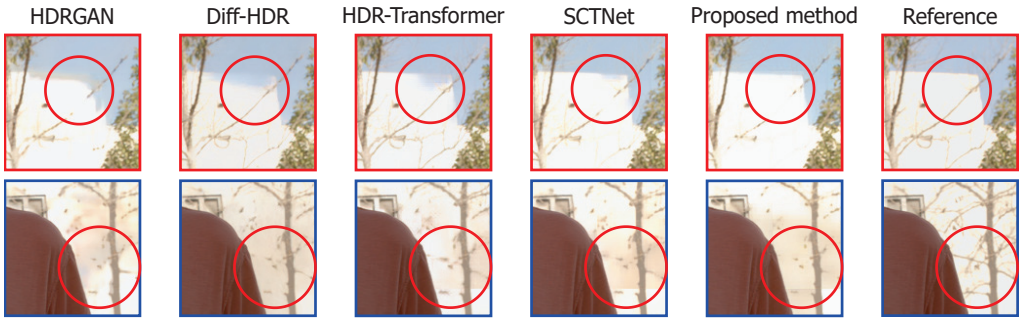
Finally, scene 7 from the Kalantari dataset is analyzed, as shown in Figure 7a, where the regions of interest are marked in red and blue. Visual results highlight the challenge of preserving features in a scenario with a varied background, where it is necessary to maintain the attributes of the tree branches over the building and the outlines of the building edges with the sky. Given this, it can be verified that in the image recovery shown in the circle within the red-marked region in Figure 7b, HDR-GAN and Diff-HDR face difficulties in reconstructing the building edges. HDR-Transformer and SCTNet exhibit the same behavior but to a lesser extent. On the other hand, the proposed method successfully addresses this problem by coherently recovering the building edges. However, when observing the tree branches, the models face difficulties in their reconstruction, as can be corroborated in the circles within the blue-marked region, which may characterize a challenge in the reconstruction process for such scenarios.

4.2.3. Comparison of Computational Cost

A comparative analysis of computational cost was conducted, with metrics including the number of parameters, the number of multiply—accumulate (MAC) units, the size of the trained model, and the inference time. As Diff-HDR is executed on a GPU, the results for all architectures were calculated using an Nvidia HGX A100 40 GB GPU. The results obtained by the proposed methodology were compared to results achieved by other methodologies analyzed, as presented in Table 3.



(**a**)



(**b**)

Figure 7. Qualitative comparison of the reconstruction of scene 7 from the Kalantari dataset [39]. (**a**) Input images with different exposures and the reference HDR image, with red and blue boxes highlighting the analyzed regions. (**b**) HDR image reconstructions by HDR-GAN [30], Diff-HDR [29], HDR-Transformer [31], SCTNet [32], and the proposed methodology, followed by the reference image. The red circles highlight the main challenges in the reconstructed images.

Table 3. Comparison of computational cost between the proposed architecture and state-of-the-art models. Results were calculated using an Nvidia HGX A100 40 GB GPU.

Method	Par. (M) ↓	MAC (G) ↓	Size (MB) ↓	Inf. (s) ↓
HDR-GAN [30]	2.63	778.61	<u>10.6</u>	**0.19**
Diff-HDR [29]	75.13	—	289.0	178.16
HDR-Transformer [31]	1.22	981.81	53.4	7.61
SCTNet [32]	<u>0.99</u>	<u>255.54</u>	28.0	6.28
Proposed method	**0.29**	**84.95**	**7.9**	<u>2.19</u>

Note: The best result is shown in **bold**, and the second best is <u>underlined</u>. ↓ indicates that lower values are better.

The results presented in Table 3 demonstrate that the proposed architecture is significantly lighter than the others, as it contains only one-third of the parameters of SCTNet, which ranks as the second-best model, reducing the number of parameters by 0.7 million. Additionally, the proposed method requires three times fewer MAC operations for reconstruction, resulting in a reduction of 170.59 GMAC compared to SCTNet, the second fastest. Similarly, the proposed method has a more compact model size than the second-smallest model, HDR-GAN, by 20.1 MB. Regarding inference time, the proposed approach achieves lower latency than most of the other architectures, being surpassed only by HDR-GAN. Consequently, the results indicate that the proposed model significantly improves computational cost compared to other state-of-the-art methodologies, as supported by an inference time that is 4.09 s faster than that of the SCTNet model.

4.3. Ablation Study

The importance of the blocks used in the proposed architecture in this paper was demonstrated through an ablation study. For this study, the architecture was divided into four key components: baseline, CBAM, SCT, and skip connection. This study helps to assess the contribution of each component to the overall performance and efficiency of the model, enabling a better understanding of the impact of each block on the HDR reconstruction task.

Baseline: Considering Figure 1, this component refers to all the convolutional blocks used in the architecture, with the exception of the convolution employed in the Skip connection.

CBAM: These blocks are used to apply channel attention and spatial attention to the input frames. They highlight the important regions within the frames to enhance feature extraction from these areas.

SCT: It combines two key components: G-SAB, which applies Transformer self-attention, and S-CAB, which applies cross-attention. Together, these components enhance the features extracted from the input frames.

Skip connection: This component involves the addition of the central frame features with those refined by the Transformer blocks, serving to retain the central image as a reference before decoding the features.

To highlight the importance of the complete architecture, a training was first conducted using only the baseline. The CBAM block was then added, followed by a second training. Subsequently, the SCT block was incorporated, and a third and final training was performed. The results for PSNR and SSIM from these trainings were calculated and are presented in Table 4.

Table 4. Results of an ablation study of the proposed architecture in terms of HDR reconstruction quality using the Kalantari dataset.

Base	CBAM	SCT	Skip c.	μ-PSNR ↑	l-PSNR ↑	μ-SSIM ↑	l-SSIM ↑
✔	✘	✘	✘	27.05	28.68	0.9518	0.9440
✔	✔	✘	✘	36.16	36.30	0.9818	0.9777
✔	✔	✔	✘	44.03	41.97	0.9915	0.9880
✔	✔	✔	✔	**44.10**	**42.07**	**0.9917**	**0.9886**

Note: ✔ means the block is present and ✘ means the block was removed from the architecture. The best result is in **bold**. ↑ indicates that higher values are better.

The results suggest that each block has a positive impact on the proposed HDR reconstruction architecture, improving the outcomes obtained in each metric, with the complete architecture achieving the best reconstruction quality. The components with the greatest impact on the architecture are CBAM and SCT, which significantly enhance the results, and the skip connection also has a positive impact, albeit to a lesser extent.

5. Conclusions

This paper proposes a Transformer-based approach for HDR reconstruction, focusing on achieving high-quality images with low computational complexity. The methodology incorporates a reduced number of self-attention blocks for feature refinement, enhancing them in an initial stage using the CBAM attention module. For comparison and validation purposes, extensive experiments were conducted, considering both quantitative and qualitative evaluations using the Kalantari and Tel datasets. Based on a quantitative comparison of the PSNR, SSIM, and HDR-VDP-2 metrics, the obtained results demonstrate that the proposed architecture significantly outperforms other state-of-the-art methodologies in terms of HDR reconstruction quality. Regarding the challenges of qualitative comparison, the results highlight the excellent performance of the proposed approach in reconstructing

image regions with high contrast in complex scenarios where limited information is available. Moreover, a comparison of the proposed methodology with other Transformer-based models, evaluated in terms of computational cost for the same HDR reconstruction task, reveals an improvement of 0.7 million parameters, 170.59 GMAC, 20.1 MB, and 4.09 s. Consequently, the proposed methodology achieves a reduction of approximately 65% in the computational cost required to perform the entire reconstruction process. Additionally, the significance of the complete architecture is emphasized through an ablation study of its main components. Finally, this approach presents a promising direction for the development of Transformer-based models with low computational cost and efficient performance in HDR reconstruction tasks.

Author Contributions: Conceptualization, J.L.-C., T.P. and A.B.A.; methodology, J.L.-C.; software, J.L.-C.; validation, J.L.-C., T.P. and A.B.A.; formal analysis, J.L.-C.; investigation, J.L.-C., T.P. and A.B.A.; resources, A.B.A.; data curation, J.L.-C.; writing—original draft preparation, J.L.-C. and T.P.; writing—review and editing, J.L.-C., T.P., A.B.A. and D.B.L.; visualization, J.L.-C.; supervision, A.B.A. and D.B.L.; project administration, T.P. and A.B.A.; funding acquisition, A.B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the PAVIC Laboratory, which benefited from SUFRAMA fiscal incentives under Brazilian Law No. 8387/1991.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Acknowledgments: The authors gratefully acknowledge support from the PAVIC Laboratory (Pesquisa Aplicada em Visão e Inteligência Computacional) at University of Acre, Brazil.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mann, S.; Picard, R. On being 'undigital' with digital cameras: Extending Dynamic Range by Combining Differently Exposed Pictures. In Proceedings of the IS&T's 48th Annual Conference: Society for Imaging Science and Technology, Washington, DC, USA, 7–11 May 1995; pp. 422–428.
2. Chaurasiya, R.K.; Ramakrishnan, K. High dynamic range imaging. In Proceedings of the 2013 International Conference on Communication Systems and Network Technologies, Gwalior, India, 6–8 April 2013; pp. 83–89.
3. Nayar, S.K.; Mitsunaga, T. High dynamic range imaging: Spatially varying pixel exposures. In Proceedings of the IEEE/CVPR Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA, 15 June 2000; Volume 1, pp. 472–479.
4. Tumblin, J.; Agrawal, A.; Raskar, R. Why I want a gradient camera. In Proceedings of the IEEE/CVPR Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 103–110.
5. Heo, Y.S.; Lee, K.M.; Lee, S.U.; Moon, Y.; Cha, J. Ghost-free high dynamic range imaging. In *Computer Vision—ACCV 2010, 10th Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010, Revised Selected Papers, Part IV*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 486–500.
6. Yan, Q.; Sun, J.; Li, H.; Zhu, Y.; Zhang, Y. High dynamic range imaging by sparse representation. *Neurocomputing* **2017**, *269*, 160–169. [CrossRef]
7. Gallo, O.; Gelfandz, N.; Chen, W.C.; Tico, M.; Pulli, K. Artifact-free high dynamic range imaging. In Proceedings of the 2009 IEEE International Conference on Computational Photography (ICCP), San Francisco, CA, USA, 16–17 April 2009; pp. 1–7.
8. Grosch, T. Fast and robust high dynamic range image generation with camera and object movement. In Proceedings of the 11th international Workshop Vision, Modeling, and Visualization, Aanchen, Germany, 22–24 November 2006; pp. 277–284.
9. Ward, G. Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. *J. Graph. Tools* **2003**, *8*, 17–30. [CrossRef]
10. Tomaszewska, A.; Mantiuk, R. Image Registration for Multi-Exposure High Dynamic Range Image Acquisition. In Proceedings of the WSCG 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Bory, Czech Republic, 29 January–1 February 2007; pp. 49–56.

11. Bogoni, L. Extending dynamic range of monochrome and color images through fusion. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR-2000), Barcelona, Spain, 3–7 September 2000; Voume 3, pp. 7–12.
12. Zimmer, H.; Bruhn, A.; Weickert, J. Freehand HDR imaging of moving scenes with simultaneous resolution enhancement. *Comput. Graph. Forum* **2011**, *30*, 405–414. [CrossRef]
13. Sen, P.; Kalantari, N.K.; Yaesoubi, M.; Darabi, S.; Goldman, D.B.; Shechtman, E. Robust patch-based HDR reconstruction of dynamic scenes. *ACM Trans. Graph.* **2012**, *31*, 203. [CrossRef]
14. Hu, J.; Gallo, O.; Pulli, K.; Sun, X. HDR deghosting: How to deal with saturation? In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1163–1170.
15. Zheng, J.; Li, Z.; Zhu, Z.; Wu, S.; Rahardja, S. Hybrid patching for a sequence of differently exposed images with moving objects. *IEEE Trans. Image Process.* **2013**, *22*, 5190–5201. [CrossRef]
16. Banterle, F.; Ledda, P.; Debattista, K.; Chalmers, A. Inverse tone mapping. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, Kuala Lumpur, Malaysia, 29 November–2 December 2006; pp. 349–356.
17. Chen, X.; Liu, Y.; Zhang, Z.; Qiao, Y.; Dong, C. Hdrunet: Single image HDR reconstruction with denoising and dequantization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 354–363.
18. Debevec, P.E.; Malik, J. Recovering high dynamic range radiance maps from photographs. In *Seminal Graphics Papers: Pushing the Boundaries*; Association for Computing Machinery: New York, NY, USA, 2023; Volume 2, pp. 643–652.
19. Liu, M.; Rehman, S.; Tang, X.; Gu, K.; Fan, Q.; Chen, D.; Ma, W. Methodologies for improving HDR efficiency. *Front. Genet.* **2019**, *9*, 691. [CrossRef]
20. Karađuzović-Hadžiabdić, K.; Telalović, J.H.; Mantiuk, R.K. Assessment of multi-exposure HDR image deghosting methods. *Comput. Graph.* **2017**, *63*, 1–17. [CrossRef]
21. Catley-Chandar, S.; Tanay, T.; Vandroux, L.; Leonardis, A.; Slabaugh, G.; Pérez-Pellitero, E. Flexhdr: Modeling alignment and exposure uncertainties for flexible HDR imaging. *IEEE Trans. Image Process.* **2022**, *31*, 5923–5935. [CrossRef]
22. Peng, F.; Zhang, M.; Lai, S.; Tan, H.; Yan, S. Deep HDR reconstruction of dynamic scenes. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 347–351.
23. Nemoto, H.; Korshunov, P.; Hanhart, P.; Ebrahimi, T. Visual attention in LDR and HDR images. In Proceedings of the 9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM), Chandler, AZ, USA, 5–6 February 2015; pp. 1–6.
24. Yoon, H.; Uddin, S.N.; Jung, Y.J. Multi-scale attention-guided non-local network for HDR image reconstruction. *Sensors* **2022**, *22*, 7044. [CrossRef]
25. Deng, Y.; Liu, Q.; Ikenaga, T. Multi-scale contextual attention based HDR reconstruction of dynamic scenes. In Proceedings of the Twelfth International Conference on Digital Image Processing (ICDIP 2020), Osaka, Japan, 19–22 May 2020; Volume 11519, pp. 413–419.
26. KS, G.R.; Biswas, A.; Patel, M.S.; Prasad, B.P. Deep multi-stage learning for HDR with large object motions. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4714–4718.
27. Liu, Z.; Lin, W.; Li, X.; Rao, Q.; Jiang, T.; Han, M.; Fan, H.; Sun, J.; Liu, S. ADNet: Attention-guided deformable convolutional network for high dynamic range imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 463–470.
28. Kong, L.; Li, B.; Xiong, Y.; Zhang, H.; Gu, H.; Chen, J. SAFNet: Selective Alignment Fusion Network for Efficient HDR Imaging. In *Computer Vision—ECCV 2024, 18th European Conference, Milan, Italy, 29 September–4 October 2024, Proceedings, Part XXVI*; Springer: Cham, Switzerland, 2025; pp. 256–273.
29. Yan, Q.; Hu, T.; Sun, Y.; Tang, H.; Zhu, Y.; Dong, W.; Van Gool, L.; Zhang, Y. Towards high-quality HDR deghosting with conditional diffusion models. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 4011–4026 [CrossRef]
30. Niu, Y.; Wu, J.; Liu, W.; Guo, W.; Lau, R.W. Hdr-gan: Hdr image reconstruction from multi-exposed ldr images with large motions. *IEEE Trans. Image Process.* **2021**, *30*, 3885–3896. [CrossRef] [PubMed]
31. Liu, Z.; Wang, Y.; Zeng, B.; Liu, S. Ghost-free high dynamic range imaging with context-aware transformer. In *Computer Vision—ECCV, 17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XIX*; Springer: Cham, Switzerland, 2022; pp. 344–360.
32. Tel, S.; Wu, Z.; Zhang, Y.; Heyrman, B.; Demonceaux, C.; Timofte, R.; Ginhac, D. Alignment-free HDR deghosting with semantics consistent transformer. In Proceeding of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 12790–12799.
33. Hittawe, M.M.; Harrou, F.; Togou, M.A.; Sun, Y.; Knio, O. Time-series weather prediction in the Red sea using ensemble transformers. *Appl. Soft Comput.* **2024**, *164*, 111926. [CrossRef]
34. Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.

35. Vaswani, A. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
36. Jagalingam, P.; Hegde, A.V. A review of quality metrics for fused image. *Aquat. Procedia* **2015**, *4*, 133–142. [CrossRef]
37. Pérez-Pellitero, E.; Catley-Chandar, S.; Shaw, R.; Leonardis, A.; Timofte, R.; Zhang, Z.; Liu, C.; Peng, Y.; Lin, Y.; Yu, G.; et al. NTIRE 2022 challenge on high dynamic range imaging: Methods and results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 1009–1023.
38. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
39. Kalantari, N.K.; Ramamoorthi, R. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.* **2017**, *36*, 144. [CrossRef]
40. Paul, N.; Chung, C. Application of HDR algorithms to solve direct sunlight problems when autonomous vehicles using machine vision systems are driving into sun. *Comput. Ind.* **2018**, *98*, 192–196. [CrossRef]
41. Kim, J.; Zhu, Z.; Bau, T.; Liu, C. DCDR-UNet: Deformable Convolution Based Detail Restoration via U-shape Network for Single Image HDR Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–18 June 2024; pp. 5909–5918.
42. Tursun, O.T.; Akyüz, A.O.; Erdem, A.; Erdem, E. The state of the art in HDR deghosting: A survey and evaluation. *Comput. Graph. Forum* **2015**, *34*, 683–707. [CrossRef]
43. Zhang, X.; Hu, T.; He, J.; Yan, Q. Efficient content reconstruction for high dynamic range imaging. In Proceedings of the ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 7660–7664.
44. Tel, S.; Heyrman, B.; Ginhac, D. CEN-HDR: Computationally efficient neural network for real-time high dynamic range imaging. In *Computer Vision—ECCV Workshops, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part II*; Springer: Cham, Switzerland, 2022; pp. 378–394.
45. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. Swinir: Image restoration using swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844.
46. Kim, J.; Kim, M.H. Joint demosaicing and deghosting of time-varying exposures for single-shot HDR imaging. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 12292–12301.
47. Zhou, F.; Fu, Z.; Zhang, D. High dynamic range imaging with context-aware transformer. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–8.
48. Chen, R.; Zheng, B.; Zhang, H.; Chen, Q.; Yan, C.; Slabaugh, G.; Yuan, S. Improving dynamic HDR imaging with fusion transformer. *AAAI Conf. Artif. Intell.* **2023**, *37*, 340–349. [CrossRef]
49. Liu, X.; Li, A.; Wu, Z.; Du, Y.; Zhang, L.; Zhang, Y.; Timofte, R.; Zhu, C. PASTA: Towards Flexible and Efficient HDR Imaging Via Progressively Aggregated Spatio-Temporal Aligment. *arXiv* **2024**, arXiv:2403.10376.
50. Yan, Q.; Chen, W.; Zhang, S.; Zhu, Y.; Sun, J.; Zhang, Y. A unified HDR imaging method with pixel and patch level. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22211–22220.
51. Chi, Y.; Zhang, X.; Chan, S.H. Hdr imaging with spatially varying signal-to-noise ratios. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5724–5734.
52. Yan, Q.; Zhang, S.; Chen, W.; Tang, H.; Zhu, Y.; Sun, J.; Van Gool, L.; Zhang, Y. Smae: Few-shot learning for HDR deghosting with saturation-aware masked autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5775–5784.
53. Wu, S.; Xu, J.; Tai, Y.W.; Tang, C.K. Deep High Dynamic Range Imaging with Large Foreground Motions. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 120–135.
54. Yan, Q.; Zhang, L.; Liu, Y.; Zhu, Y.; Sun, J.; Shi, Q.; Zhang, Y. Deep HDR imaging via a non-local network. *IEEE Trans. Image Process.* **2020**, *29*, 4308–4322. [CrossRef] [PubMed]
55. Yan, Q.; Gong, D.; Shi, Q.; Hengel, A.v.d.; Shen, C.; Reid, I.; Zhang, Y. Attention-guided network for ghost-free high dynamic range imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1751–1760.
56. Chen, J.; Yang, Z.; Chan, T.N.; Li, H.; Hou, J.; Chau, L.P. Attention-guided progressive neural texture fusion for high dynamic range image restoration. *IEEE Trans. Image Process.* **2022**, *31*, 2661–2672. [CrossRef] [PubMed]
57. Song, J.W.; Park, Y.I.; Kong, K.; Kwak, J.; Kang, S.J. Selective transhdr: Transformer-based selective HDR imaging using ghost region mask. In *Computer Vision—ECCV, 17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XVII*; Springer: Cham, Switzerland, 2022; pp. 288–304.
58. Jam, J.; Kendrick, C.; Drouard, V.; Walker, K.; Hsu, G.S.; Yap, M.H. R-mnet: A perceptual adversarial network for image inpainting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 2714–2723.

59. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801. [CrossRef]
60. Sara, U.; Akter, M.; Uddin, M.S. Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. *J. Comput. Commun.* **2019**, *7*, 8–18. [CrossRef]
61. Mantiuk, R.; Kim, K.J.; Rempel, A.G.; Heidrich, W. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph. (TOG)* **2011**, *30*, 1–14. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

MixRformer: Dual-Branch Network for Underwater Image Enhancement in Wavelet Domain

Jie Li, Lei Zhao *, Heng Li, Xiaojun Xue and Hui Liu

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China; lijie515@stu.kust.edu.cn or 20222204196@stu.kust.edu.cn (J.L.); liheng@kust.edu.cn (H.L.); xiaojun35678@kust.edu.cn (X.X.); liuhui621@126.com (H.L.)
* Correspondence: zhaolei@kust.edu.cn

Abstract: This paper proposes an underwater image enhancement model MixRformer that combines the wavelet transform and a hybrid architecture. To address the problems of insufficient global modeling in existing CNN models, weak local feature extraction of Transformer and high computational complexity, multi-resolution feature decomposition is performed through a discrete wavelet transform (IWT/DWT) in which low-frequency components retain structure and texture, and high-frequency components capture detail features. An innovative dual-branch feature capture module (DFCB) is designed as follows: (1) the surface information extraction block combines convolution and position encoding to enhance local modeling; (2) the rectangular window gated Transformer expands the receptive field through the convolution gating mechanism to achieve efficient global relationship modeling. Experiments show that the model outperforms mainstream methods in color restoration and detail enhancement, while optimizing computational efficiency.

Keywords: underwater image enhancement; wavelet transform; convolutional neural network; Transformer; multi-branch structure

1. Introduction

As an information carrier for understanding and perceiving the underwater environment, underwater images play an important role in marine resource exploration, marine biodiversity research, underwater rescue, target identification, real-time tracking, etc. However, underwater images are often affected by light absorption and scattering caused by particles in the water, which can significantly reduce the color and contrast of underwater images. In addition, light of different wavelengths can also affect the quality of imaging. Due to the poor underwater imaging environment, color casts, color artifacts, and blurred details are common, even in high-end cameras. Underwater image enhancement technology was invented to improve the quality of underwater images. Therefore, the use of underwater image enhancement technology to obtain clear underwater images plays a vital role in various fields such as scientific research and resource development. Existing underwater image enhancement methods are generally divided into three categories: methods based on traditional physical models, methods based on visual priors, and methods based on deep learning.

Traditional physical models can enhance underwater images of specific scenes. He, K., Sun, J., & Tang, X et al. [1] proposed a single-image dehazing method based on dark channel prior, which effectively removed the haze of underwater images by simulating the atmospheric scattering model. Drews, P., Nascimento, et al. [2] proposed an algorithm for estimating underwater image transmission using the atmospheric scattering model,

which effectively improved the clarity and contrast of the image. Although the physical model-based method can restore the appearance of underwater images, it relies on specific parameter estimation and cannot distinguish between texture detail information and noise in the image. While improving image details, it also inevitably increases noise. For images with uneven lighting, fog, high noise, etc., local areas are prone to over-enhancement, while background areas are prone to under-enhancement. The visual prior-based method ignores the physical model and improves the image visual quality by adjusting the image pixel values of contrast, brightness and saturation to comprehensively evaluate the image quality. Zhang et al. [3] proposed an enhancement method based on Retinex theory. By applying filters to each component in the Lab color space, they successfully eliminated noise and halo artifacts in the image, significantly improving the image quality. Drews, P., Nascimento, E., et al. [4] proposed an image restoration method based on depth estimation, which removes turbidity and improves image quality by estimating the depth of underwater scenes. Although it has strong adaptability and robustness and can provide consistent results in complex environments, its effect may be limited by specific prior knowledge or statistical characteristics, and may not perform as expected in low-contrast or high-noise images, resulting in artifacts, color deviation, oversaturation, and other problems in the generated images.

In recent years, more and more researchers have begun to build different deep learning model structures and have achieved good results and high efficiency. Most of the underwater image enhancement methods based on deep learning [5] are centered on end-to-end models and use pairs of degraded underwater images and high-quality corresponding images to train the model to effectively enhance the image contrast and correct the image color. However, after a large number of experiments on challenging datasets, we found that some methods often lead to blurred images or introduce unrealistic color changes. This is because the model ignores certain information under its powerful correction ability. Convolutional neural network [6] is the most classic network in computer vision. Underwater images are enhanced using a convolutional neural network. Although convolutional networks can effectively achieve high-precision local enhancement, it is difficult for convolution to model global long-range relationships, which seriously limits the development of underwater image enhancement tasks. To address the above problems, some articles use the self-attention module of Transformer [7] to expand the model's ability to capture global long-range relationships. However, although Transformer has shown excellent performance in visual tasks, it lacks effective inductive bias and cannot encode the two-dimensional position information of the image. It needs to embed relative or absolute position encoding, which may lead to its encoding and generalization capabilities being inferior to convolutional networks. We believe that there are four reasons for this weakening. First, Transformer can effectively handle the dependencies between tokens, but its complexity is quadratic with the resolution of the image, that is, $O((HW))^2$, which can easily lead to insufficient GPU memory, low computational efficiency, and long training time. Secondly, due to the lack of interaction between windows in the square window of Transformer, the receptive field increases slowly, which affects the capture and processing of global information. In addition, the traditional feedforward layer MLP in Transformer is mainly used to process the information interaction of feature dimensions. Although the structure is simple and universal, the existence of GELU will truncate some data, resulting in the loss of key information, and more importantly, MLP cannot actively select which information needs to be retained or suppressed. Finally, Transformer mainly relies on global attention, cannot encode the two-dimensional position information of the image, and is difficult to model local information, resulting in a lack of local receptive fields and insufficient attention to details.

In order to solve the above problems, this paper proposes a dual-branch underwater image enhancement algorithm, MixRformer, that integrates CNN and Transformer in the wavelet domain. It is based on the U-Net structure. The convolutional layer of the original U-Net is replaced by a dual-branch feature extraction module (DFCB) for local and global interaction of multi-scale features and combined with jump connections to improve feature fusion and spatial information recovery.

In order to reduce the computational complexity of the Transformer and improve the inference speed, we introduce a wavelet transform (DWT-IWT) for up-sampling and down-sampling [8], which reduces the resolution while retaining key features. In addition, inspired by CAT [9], we propose the Rectangle GLU-Window Transformer module, which combines horizontal and vertical rectangular window attention mechanisms to expand the receptive field without increasing the computational complexity and enhances the information interaction between windows through Axial-Shift. The feedforward network is also crucial for the Transformer. We use ConvGLU instead of MLP, introduce a gating mechanism to dynamically adjust channel activation, selectively retain or suppress information, and use convolution instead of SE modules to enhance local feature modeling ability. Given that images are different from sequence data and have a two-dimensional spatial structure, the Transformer is less efficient in extracting local features (such as edges, textures) and encoding two-dimensional information. Compared with the recent CNN-Transformer model, we first introduced the wavelet transform into a dual-branch structure for underwater enhancement tasks. At the same time, on the basis of reducing the computational complexity and parameter quantity, we combined the frequency domain information to synergistically improve various indicators of underwater images. Therefore, combining the advantages of CNN and Transformer is an effective way to improve the image enhancement effect.

In general, the unique contributions of this paper are summarized as follows:

1. This work proposes, for the first time, an underwater image enhancement algorithm structure, MixRformer, that effectively combines the wavelet transform with CNN-Transformer. The introduction of the wavelet transform is not only conducive to restoring the detailed features of the image, but also can reduce the image resolution, thereby reducing GPU memory consumption and reducing the amount of calculation of Transformer, so that MixRformer can better restore the color and texture of the image.
2. We propose a dual-branch feature capture block (DFCB), which consists of a simple surface information extraction block, ConvBlock, and an innovative Rectangle GLU-Window Transformer block, which are used to extract surface details and capture global features, respectively.
3. We construct a multi-loss function and introduce MSE Loss and VGG Perceptual Loss to train the model for restoring underwater distorted images. Compared with several state-of-the-art underwater image enhancement (UIE) methods, our proposed method shows extremely excellent performance in both objective indicators and visual quality. In particular, it has significant advantages in eliminating color casts and improving image clarity.

2. Related Works

2.1. Underwater Image Restoration Task

Underwater image restoration is a long-standing low-level vision problem that aims to restore high-quality images from low-quality images. In recent decades, many researchers have proposed various theories and methods to improve the quality of image restoration in order to solve these challenging tasks. For example, 1. Hou et al. proposed a variational model based on non-local micromolecules, which effectively improved the dehazing,

denoising and visibility of underwater images [10]. 2. Zhou et al. proposed a method based on a color line model to generate underwater images with true color and natural appearance by filtering out image blocks with color line prior features [11]. 3. Song et al. proposed a systematic underwater image restoration method that estimates underwater light through quadtree refinement and optimizes the transmission map to enhance contrast [12]. 4. Li et al. proposed an underwater image enhancement algorithm based on an improved dark channel prior algorithm to solve the problem of too dark or too bright areas in the image [13]. The above methods mainly use underwater image restoration technology to focus on inversely solving the image degradation problem caused by light absorption and scattering in the underwater environment, thereby restoring the original clarity and color information of the image. However, these methods can only achieve satisfactory results when the underwater scene conforms to the selected model. Although the goal is clear, it limits the robustness and scalability of the model in complex and changing situations. In contrast, underwater image enhancement is more concerned with improving the observability of the image, focusing on processing the degraded image itself, and can retain and emphasize the visual details, color information, texture information and overall structure in the image, making the image more informative and visually effective, and has strong generalization ability and can handle diverse underwater environments. In summary, in order to improve the visual effect of underwater images, we propose a novel underwater image enhancement method.

2.2. CNN-Based UIE Methods

With the rapid development of convolutional neural networks, image restoration methods based on convolutional neural networks have achieved certain results, and more and more researchers have begun to apply convolutional neural networks to underwater image enhancement. The success of these methods is attributed to their powerful feature extraction capabilities and carefully designed network architectures, which can extract coarse-grained and fine-grained features in different receptive fields. For example, Anwar et al. [14] proposed the UWCNN network, which effectively improved the denoising and reconstruction performance of low-quality images by introducing a combination of super-resolution and convolutional neural networks. They also constructed a comprehensive underwater image dataset based on the physical model of underwater imaging and different water quality conditions for training the proposed underwater image convolutional neural network. In addition, Chen et al. [5] proposed an underwater image enhancement method based on deep convolutional neural networks, which effectively restored the color and details of the image by learning the local and global features of underwater images.

In order to better encode the global information of the image, the current research goal is to explore more powerful deep learning models. Most convolutional neural networks (CNNs) are good at extracting local features, and underwater images are enhanced using convolutional neural networks. Although convolutional neural networks can effectively achieve high accuracy, it is difficult for convolution to model global long-range relationships, which seriously limits the development of underwater image enhancement tasks. In contrast, the recently emerging Transformer model has shown significant advantages in capturing long-range dependencies and is therefore widely considered to be an effective way to improve the performance of image processing and enhancement tasks.

2.3. Transformer-Based UIE Methods

In addition to the traditional CNN architecture, in recent years, Transformer-based architectures have also been gradually applied to underwater image enhancement tasks. Transformer has shown significant advantages in capturing long-range dependencies and

modeling global information, and has, therefore, become an important technology for enhancing underwater images. For example, as a new type of visual Transformer, Swin Transformer adopts a windowed self-attention mechanism, which can effectively improve computational efficiency while retaining global information, which has obvious advantages in underwater image enhancement tasks [15]. In addition, the UNet architecture has been widely used in underwater image enhancement. UNet can efficiently extract local features of images and maintain spatial information through a symmetrical encoding-decoding structure. The U-TransCNN model proposed by Y et al. [16] introduces the Transformer module into the encoder of UNet, making full use of the local feature extraction capability of CNN and the global information modeling advantage of Transformer, significantly improving the restoration effect of underwater images. Chen et al. [17] also proposed a hybrid model combining Transformer and convolutional neural networks to process information in underwater images. This method effectively improves the enhancement effect of underwater images under different water conditions by combining convolution and self-attention mechanisms. Although Transformer can improve the performance of the model, its own mechanism will bring a lot of GPU memory consumption and time overhead.

Therefore, our goal is to design a novel and more efficient Transformer for underwater image restoration based on the integration of CNN. At present, the dual-branch structure that can retain local features and global representation to the greatest extent is still temporarily shelved. The Dual-branch Feature Capture Block (DFCB) module is not only integrated into our MixRformer model through the dual-branch structure, but the MixRformer also pays more attention to details and global features and increases the model receptive field while keeping the computational complexity unchanged, reducing the number of model parameters. DFCB will be explained in detail in Section 3.3.

2.4. Wavelet-Based UIE Methods

Wavelets are widely used in underwater image enhancement (UIE) tasks. With the rise of deep learning, some studies have combined wavelets with CNN and achieved excellent results. For example, Wang et al. [18] designed a multi-level wavelet-based underwater image enhancement network (MWEN), combining a multi-level wavelet transform with color compensation prior to enhance underwater images. Subsequently, Zhao et al. [19] proposed a wavelet–Fourier information interaction with frequency diffusion adjustment (WFII-FDA), introducing a wavelet–Fourier information interaction module and a frequency residual diffusion adjustment module to enhance the high-frequency and low-frequency components of underwater images. Wang et al. [20] introduced a wavelet pixel domain progressive fusion network, which includes a wavelet domain module (WDM) to extract multi-scale frequency features with fine-grained details, thereby improving the quality of underwater images. Inspired by these methods, we intend to explore the performance of Transformers in the wavelet domain and build a more lightweight Transformer model using wavelets.

2.5. Style and Keypoint Learning-Based UIE Methods

In the field of underwater image processing, in addition to direct enhancement and feature extraction methods, style contrast learning and keypoint-sensitive learning provide important perspectives for related tasks. Recently, Wang et al. (2025) [21] proposed the content style control network (CSC-SCL), which is based on the CycleGAN framework. It separates content and style features through content control normalization (CCNorm) and style control normalization (SCNorm) and combines style contrast learning to improve the quality of underwater image enhancement, achieving higher indicators on the UIEB

dataset. This method performs well in improving visual quality and domain generalization. On the other hand, Chen et al. (2024) [22] proposed hierarchical progressive learning (HTPL-JB), which optimizes feature extraction in sonar image classification through jigsaw self-supervised learning, hierarchical pre-training, and progressive training based on block convolution, combined with key point sensitive loss (KPSLoss), solves the problems of few samples and long-tail distribution, and achieves a G_mean of 94.77% on the NKSID dataset. Although the two methods focus on enhancement and classification, respectively, which are different from the underwater image enhancement task of this study (based on wavelet decomposition and dual-branch Transformer-CNN architecture), their feature processing strategies provide useful ideas for this field.

3. Proposed Approach

In this section, we first present the details of the proposed MixRformer model architecture, which is designed for underwater image enhancement with good robustness. Next, we elaborate on the key components of the network and finally introduce the loss function used for training.

3.1. Network Architecture

Our main goal is to develop a model that combines the characteristics of Transformer and CNN for underwater image enhancement in the wavelet transform domain by effectively co-optimizing the low- and high-frequency features in the wavelet transform domain. In computer vision tasks, the low frequencies in the wavelet transform domain mainly contain the main structure and basic texture information of the image, while the high frequencies correspond to the edges and detail features of the image. Low- and high-frequency information plays an important role in image restoration. Transformer and convolutional neural networks (CNNs) each have their own advantages. Transformer is good at capturing global dependencies and long-range relationships, but its computational complexity is high, which limits its application in high-resolution images. On the contrary, the CNN efficiently extracts detail features through local convolution operations, with low computational overhead, and is particularly suitable for processing edges and textures in images.

Therefore, combining Transformer with CNN can take advantage of the advantages of both: Transformer captures global information, and CNN extracts surface features. This combination not only improves the performance of the model, but also optimizes computational efficiency. As shown in Figure 1, the model MixRformer proposed in this study mainly consists of four parts: discrete wavelet transform (DWT), dual-branch feature extraction module (DFCB), inverse wavelet transform (IWT) and coarse skip-connection. Specifically, the model is built on a U-shaped architecture [23]. First, because the resolution of the input image is large, the model downsamples the input image through DWT, effectively extracting the high and low frequency information of the image while reducing the image resolution.

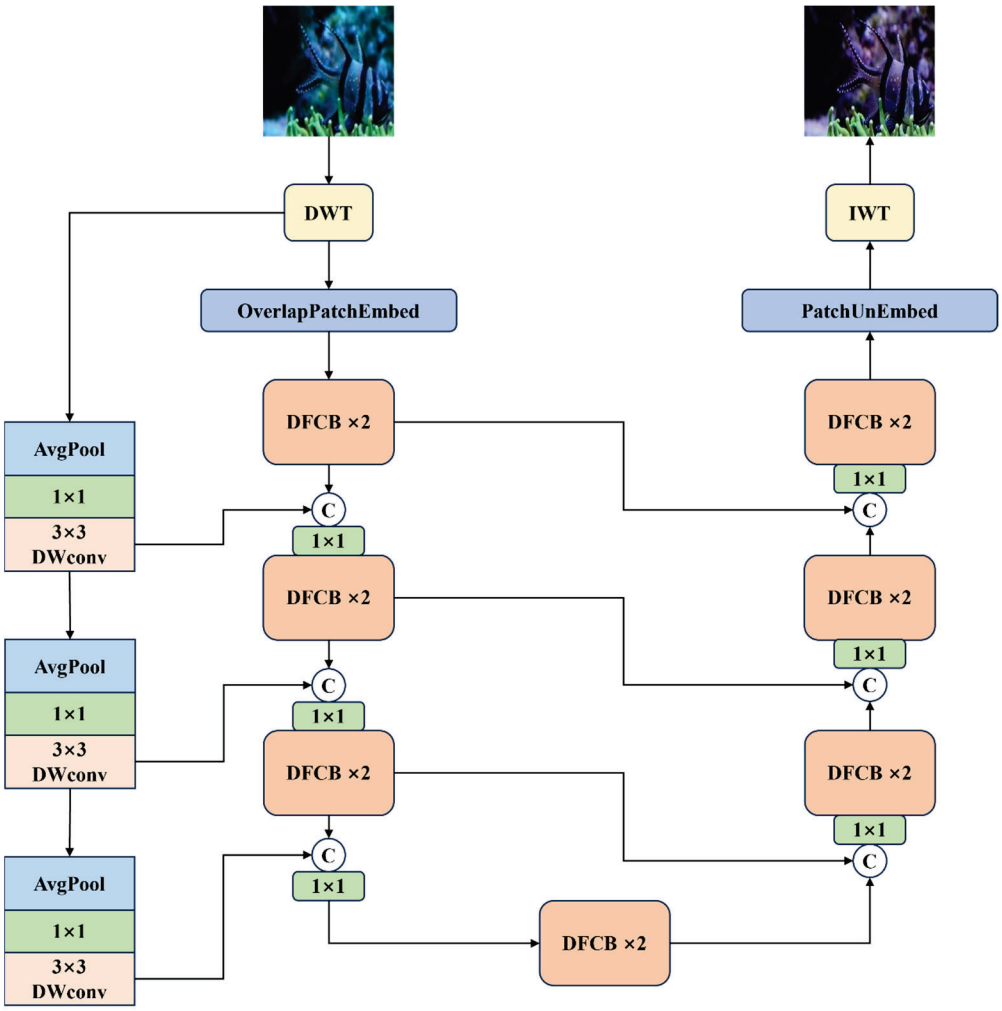


Figure 1. The overall architecture of MixRformer. It mainly combines wavelet transform with DFCB and coarse skip connections to process the information of underwater images in the frequency domain. The components will be introduced one by one later.

First, due to the large resolution of the input image, the model samples the input image through DWT, effectively extracting the high-frequency and low-frequency information of the image while reducing the image resolution, and concatenates the low-frequency and high-frequency four frequency domain image information to form a new input feature map, so that the feature map contains low-frequency and high-frequency information. The main part of the model uses the DFCB feature extraction module to perform feature processing on the new feature map after concatenation. We use DFCB to replace the pure convolution operation in the U-shaped structure, where the encoder gradually extracts the features of the input image through convolution layers and pooling layers and reduces the spatial resolution. On the other hand, at each stage of the encoding stage, our model sets a coarse skip connection. The role of this connection is to supplement the original features of the input data. It consists of a series of operations, including average pooling, pixel-level convolution, and depth convolution. Such a setting enables the encoder to focus more effectively on learning the residual caused by the absorption of light by water. In the encoder part, we give the network input as $I_C \in H \times W \times C$, In order to extract the wavelet domain information of the image and obtain low and high frequency information, DWT downsampling h_{DWT} converts the input picture I_C into four wavelet images:

$$I_{LL}, I_{LH}, I_{HL}, I_{HH} = h_{DWT}(I_C) \tag{1}$$

where $I_{LL}, I_{LH}, I_{HL}, I_{HH} \in \frac{H}{2} \times \frac{W}{2} \times C$ are 4 sub-images with different frequencies. Then, we connect the four sub-bands into a shallow feature $F_e \in \frac{H}{2} \times \frac{W}{2} \times 4C$. In order to fuse the low-frequency and high-frequency information, F_e is subjected to a convolution operation for adaptive learning to fuse the features of each frequency and divide them into patches. These features are sent to the encoder composed of DFCB to further extract more effective features. In order to preserve the integrity of the image and reduce distortion, each layer of the encoder is combined with the shallow features of the upper layer through a coarse skip layer. When the network enters the bottleneck part, the bottleneck layer provides richer feature support for the decoder through global feature extraction and information compression to improve the enhancement effect. This part consists of only two dual-branch feature extraction modules. The decoder part is used for image reconstruction and adopts the same structure as the encoder part, retaining the dual-branch feature extraction module. Finally, in order to restore the image to the RGB channel, the IWT operation is used to convert the features to the original resolution and reconstruct the restored image.

3.2. Wavelet-Based Image Enhancement

In the image enhancement task, the low-frequency and high-frequency information in the image is a very important key factor for image restoration. The low-frequency sub-band contains the main structure and overall contrast information of the image, while the high-frequency sub-band corresponds to details such as edges and textures. At the same time, due to the large resolution of the general input image, the computational cost of the model for processing large-resolution images will be greatly increased. In order to solve the above problems, many methods have been proposed so far, such as convolution or pooling operations. However, some of these operations will cause irreversible information loss in the image, which is not conducive to image enhancement processing. Therefore, in order to solve the above problems, we introduced the wavelet transform into the model, which reduced the resolution of the image while introducing low-frequency and high-frequency information, greatly reducing the inference time and computational cost of the model.

Multi-scale frequency domain feature extraction [24] plays an important role in image enhancement tasks. As shown in Figure 2, the input image is subjected to multi-resolution analysis using a discrete wavelet transform (DWT), which can be decomposed into four sub-band components with different spatial frequency characteristics. A large number of studies and experiments have shown that the sub-bands decomposed by wavelet carry multi-dimensional feature representations of the image: the low-frequency component (LL) retains the global structure of the original image as a basic approximate representation; the vertical high-frequency component (HL) and the horizontal high-frequency component (LH), respectively, record the edge contour features in the corresponding directions; and the diagonal high-frequency component (HH) contains important oblique texture details. Combining features of different frequencies with the deep model of the Unet architecture can effectively guide the network to focus on visual features of different scales, thereby enhancing the model's ability to restore image texture.

The innovative advantages of the wavelet transform we have made are as follows: specifically, the wavelet transform can be completely reversible for images and achieve lossless reconstruction of image information; secondly, the resolution of feature maps can be effectively reduced by multi-level the wavelet transform, which can greatly reduce the consumption of computing resources compared with traditional reduction methods, and does not introduce additional trainable parameters, significantly improving the efficiency of model reasoning; thirdly, sub-band components have dual representation capabilities of spatial position and frequency domain features, which is conducive to the network capturing geometric structure features in different directions; overall, the multi-scale

characteristics of wavelet functions can expand the receptive field of the network, and on this basis, the multi-level feature fusion mechanism is used to enhance the model's ability to express complex textures, providing an effective physical prior guidance for the model.

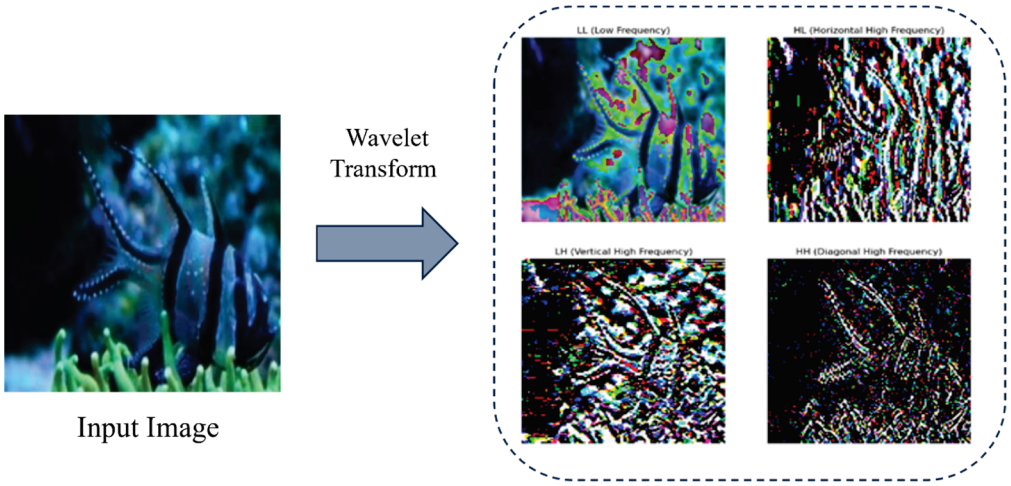


Figure 2. Schematic diagram of a wavelet transform.

3.3. Dual-Branch Feature Capture Block

According to most previous Transformer-based methods, we found that they simply use the convolutional layer network for feature aggregation or sampling. However, the computational complexity of the Transformer itself is too large, which not only consumes a lot of computing resources but also causes the model performance to decrease due to multiple stacking of Transformers.

$$Attention(Q, K, V) = Softmax\left(Norm\left(QK^T\right)\right)V \tag{2}$$

At present, CNN has been used as the backbone network in image processing. Its ability to extract local features and image location information is innate. Compared with Transformer's artificial location encoding ability, it is far from CNN's ability to automatically learn location information. However, Transformer can directly model global features through self-attention, allowing each pixel to pay attention to all pixels in the entire image. At the same time, Transformer itself can adaptively pay attention to information of different scales. Like CNN, it has its own natural multi-scale modeling ability, especially in dealing with complex details. Therefore, considering the multi-scale modeling ability, we found that introducing a multi-branch structure into the model can better extract visual features of different granularities. The multi-branch structure adopts a parallel structure to extract different features without interfering with each other. This design can avoid the problem of information sparsity caused by deep network stacking. As shown in Figure 3, our DFCB uses CNN and Transformer as two different branches, as surface information extraction branch and global information extraction branch, respectively. When shallow features are input into DFCB, they will enter both branches at the same time. Specifically, the surface information extraction branch contains a lightweight ConvBlock, which is a simple module consisting of two convolutional layers and a ReLU activation function. It introduces local residual connections to enhance feature reuse while retaining the surface information of the image. In the global information extraction branch, we designed an RGLUWin Transformer module to expand the attention receptive field without increasing the amount of computation, thereby maintaining the suppression or retention of information.

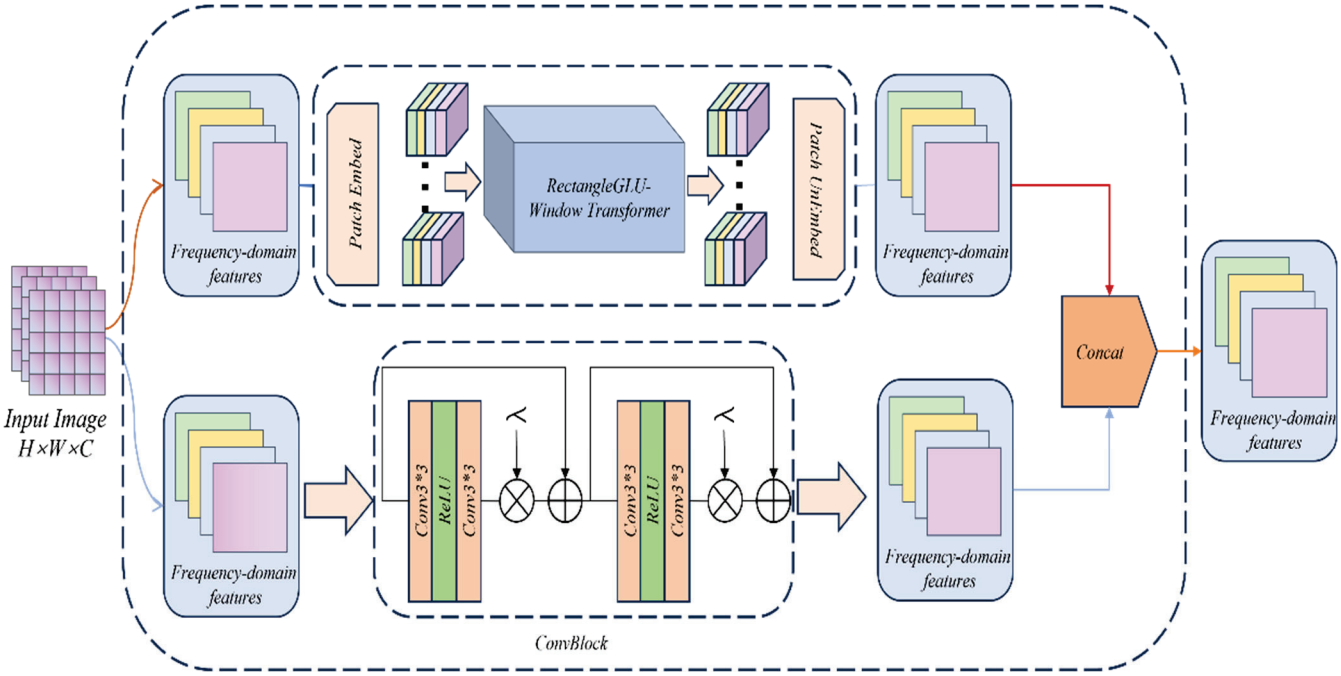


Figure 3. Detailed structure diagram and data flow of DFCB.

Rectangle GLU-Window Transformer Block

The RGLUWin Transformer module is a key part of our dual-branch feature extraction module. As shown in Figure 3, the RGLUWin Transformer is used to extract the global context information of the image. It is more suitable for image restoration than the previous Transformer structure. Because the Transformer enables all image blocks to interact through global dependency modeling capabilities, its complexity is quadratic with the resolution of the image. Although local window self-attention reduces the computational complexity, the fixed division limits the cross-window interaction, resulting in limited receptive field expansion and insufficient modeling capabilities, affecting image restoration performance. For the above problems, we adopt a rectangular window self-attention mechanism, which is divided into a horizontal rectangular window (H-Rwin) and a vertical rectangular window (V-Rwin), and used in parallel for different attention heads. By aggregating features across different windows, the attention area is expanded without increasing the computational complexity, and the rectangular window self-attention is realized. In addition, the rectangular window can capture the different features of each pixel in the horizontal and vertical directions and aggregate the features of different windows to expand the receptive field, so that the model can build a differentiated receptive field, while the traditional square window is difficult to take into account the multi-scale and multi-directional feature expression requirements. The RGLUWin Transformer is shown in the Figure 4.

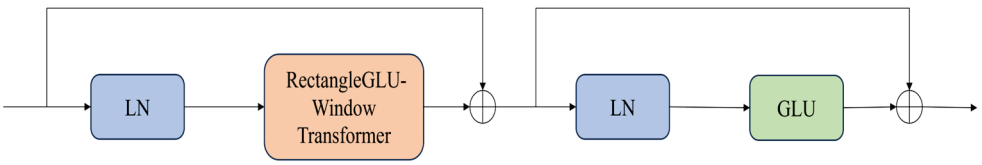


Figure 4. Block diagram of the RGLUWin Transformer.

The special RGLUWin Transformer uses a rectangular window as shown in the Figure 5. ($sh \neq sw$) instead of square windows ($sh = sw$), where sh and sw represent

the height and width of the rectangle, respectively. In addition, we divide the rectangular window into horizontal windows ($sh < sw$, denoted as H-Rwin) and vertical windows ($sh > sw$, denoted as V-Rwin), and use them for parallel computation of different attention heads. In general, M attention heads are performed in parallel on the input feature $X \in R^{H\times W\times C}$. For each attention head, $X \in R^{H\times W\times C}$ is split into non-overlapping $sh * sw$ rectangular windows, and the I-th rectangular window feature is represented as $X_i \in R^{(sh\times sw)\times C}$ where $i = 1,,, \frac{H\times W}{sh\times sw}$ then the self-attention of the m-th head can be calculated as follows:

$$\begin{aligned}(Q_i{}^m, K_i{}^m, V_i{}^m) &= \left(X_i W_m{}^Q, X_i W_m{}^K, X_i W_m{}^V\right),\\ Y_i^m &= Attention(Q_i{}^m, K_i{}^m, V_i{}^m) = SoftMax(\frac{Q_i{}^m (K_i{}^m)^T}{\sqrt{d}} + B)V_i{}^m\end{aligned} \tag{3}$$

where $Y_i^m \in R^{(sh\times sw)\times D}$ is the attention feature of X_i in the m-th head, $Q_i{}^m, K_i{}^m, V_i{}^m$ represent the projection matrices of the query, key and value of the head, respectively, $D = d = \frac{C}{M}$ is the channel dimension of each head, and B is the dynamic relative position encoding. Performing attention operations on all $X_i (i = 1,,, \frac{H\times W}{sh\times sw})$, and reshaping and merging in the partitioning order, we can obtain the attention feature $Y^m \in R^{H\times W\times D}$ of the entire X.

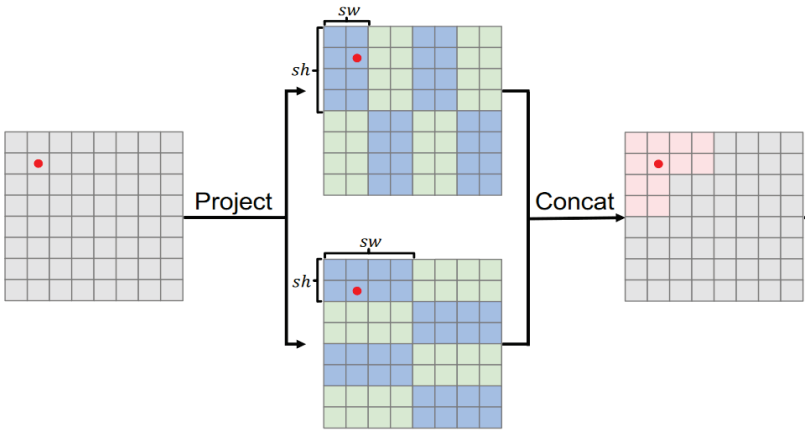


Figure 5. Rectangular Window Self-Attention Mechanism.

Assuming that the number of attention heads M is even, the attention heads are evenly divided into two parts, and H-Rwin is performed on the first part and V-Rwin is performed on the second part. The outputs of the last two parts are concatenated according to the channel dimension. The calculation formula of this process can be expressed as follows:

$$Rwin - SA(X) = Concat\left(Y^1, Y^2, \ldots, Y^M\right) W^p \tag{4}$$

Among them, $Y^1, \ldots, Y^{\frac{M}{2}}$ are output using the H-Rwin head, and $Y^{\frac{M}{2}+1}, \ldots, Y^M$ are output using the V-Rwin head. $W^p \in \mathrm{R}^{C\times C}$ represents the projection matrix of feature fusion.

At the same time, we found that the traditional MLP layer has inevitable shortcomings for underwater images. The GELU in the MLP will truncate the data, resulting in information loss. It also lacks selective information suppression or retention, and is not sufficient for position encoding. Inspired by [25], as shown in Figure 6 each token in the ConvGLU has a unique gating signal based on its most recent fine-grained feature. The gating mechanism suppresses the noise channel and strengthens the transmission of important features. At the same time, by introducing the 3×3 deep convolution, not only the shortcomings of the excessive coarse-grained global average pooling in the SE mechanism are solved, but also

the deep convolution can provide position encoding. The 3×3 deep convolution retains spatial locality and makes up for the shortcomings of the Transformer in fine-grained texture modeling. In addition, the value branch of this design still maintains the same depth as the MLP and GLU, which is friendly to its back-propagation. Its calculation expression can be expressed as follows:

$$ConvGLU(X) = (XW_1) \odot GELU(DWConv(XW_2)) \quad (5)$$

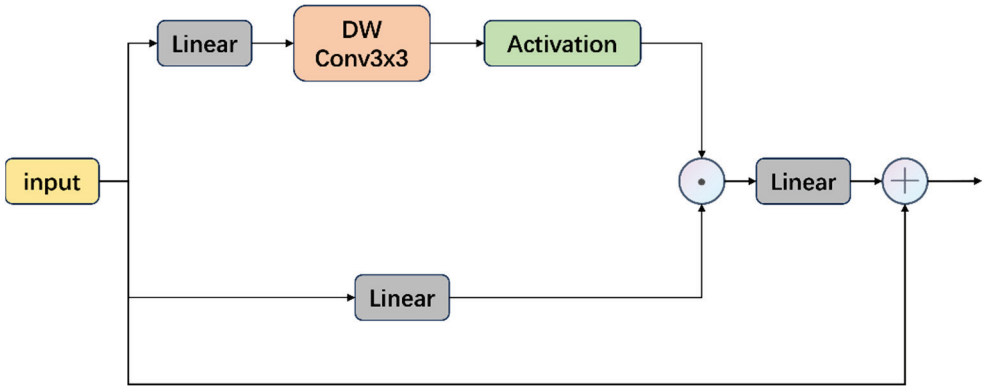


Figure 6. Detailed structure diagram of GLU.

Among them, DWConv is a depth-wise separable convolution, and ⊙ represents element-by-element multiplication.

In general, the RGLUWin Transformer block we proposed helps the model capture the global information of the image through a rectangular window, and combines it with the GLU module to enable the model to retain useful information or suppress interference information in long-dependency information.

3.4. Loss Function

To train our model, we formulate a multinomial loss function which consists of two parts.

3.4.1. MSE Loss

The mean square error loss is one of the basic functions in regression tasks. The model error is quantified by calculating the squared average of the difference between the generated image $\hat{I}_j$ and the reference image I_j. Its mathematical expression is as follows:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{I}_j - I_j\right)^2 \quad (6)$$

3.4.2. VGG Perceptual Loss

The calculation of mean square error loss is to optimize the model by the difference of pixel values. Excessive pursuit of pixel alignment may lead to overly smooth results and lack of details. Compared with mean square error loss, content-aware loss extracts high-level features through a pre-trained deep network (VGG) to measure the similarity between the generated content and the target in the feature space. In this chapter, a pre-trained 19-layer VGG network is used as a feature extractor [26], and the perceptual loss function is defined with a ReLU activation layer. Specifically, let $\phi_j(x)$ be the jth convolutional layer (after activation) of the VGG19 network ϕ pre-trained on the ImageNet dataset. The perceptual loss function is constructed by calculating the distance between the enhanced

image I_e and the reference image I_g in the feature space, and its mathematical expression is as follows:

$$L_{VGG} = L_j^{\phi} = \frac{1}{C_j H_j W_j} \sum_{i=1}^{N} \| \phi_j(I'_e) - \phi_j\left(I'_g\right) \| \tag{7}$$

where N represents the number of samples in each batch during training; $C_j H_j W_j$ represent the number, height and width of the feature map of the jth convolutional layer in the VGG19 network, which together define the dimension of the feature map.

In general, the total loss function can be expressed as follows:

$$L_{TOIAL} = \lambda_1 L_{MSE} + \lambda_2 L_{VGG} \tag{8}$$

The coefficients are the weights of each loss, and we empirically set λ_1 and λ_2 to 1 and 1.

4. Experiments

4.1. Implementation Details

4.1.1. Datasets

In this paper, to train and verify the generalization ability and effectiveness of MixRformer, we use the public underwater image dataset EUVP (https://irvlab.cs.umn.edu/resources/euvp-dataset) (accessed on 20 May 2025) [27]. This public dataset contains a large number of real underwater images and is divided into two subsets: paired data and unpaired data. The main advantages of the EUVP dataset are data diversity and strong restoration of real scenes. The images in this dataset are collected from different water environments, covering a variety of underwater environments (e.g., different turbidity, lighting conditions, water body types, etc.), and are shot using professional equipment, which better reflects the various challenges in real underwater imaging. At the same time, the dataset contains tens of thousands of images, which is very friendly to models containing Transformer and can reduce the risk of overfitting. This paper uses a paired dataset to train 11,435 images, of which 1270 images are used for verification and 515 images are used for testing. At the same time, in order to verify the generalization ability of our model, we use the 120 images in the test part of the public dataset UFO-120 (https://irvlab.cs.umn.edu/resources/ufo-120-dataset) (accessed on 20 May 2025) for generalization experiments. Experimental results show that our proposed model is superior in underwater image enhancement.

4.1.2. Experimental Settings

Our network was implemented on Ubuntu 20 with Intel (R) Xeon E7-4809 v3 CPU, NVIDIA RTX 3090 GPU and PyTorch 3.10.6 framework. The model was trained for 100 epochs. We used Adam optimizer to train the model. To quantitatively compare the quality of enhanced underwater images, we used full-reference evaluation and no-reference evaluation metrics to measure the performance of different methods. We considered two commonly used full-reference evaluation standard metrics, namely Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [28], and three commonly used no-reference evaluation metrics, namely Underwater Image Quality Metric (UIQM), Underwater Image Color Quality Evaluation Metric (UCIQE) and Natural Image Quality Evaluation Metric (NIQE) to quantify and evaluate the color and structural similarity between MixRformer enhanced images and their corresponding real images. Generally, higher values of PSNR, SSIM, UIQM and UCIQE indicate better performance, and lower values of NIQE indicate better performance.

4.2. Comparative Experiments and Analysis

To demonstrate the performance of the proposed model, in this section we compare the MixRformer model with the following ten competitive underwater image enhancement models, which cover a variety of methods, including BRUE [29], based on Retinex variational theory, UDCP [4] and IBLA [30], based on physical models, ShallowUWnet [31] and Uice2Net [32], based on convolutional neural networks (CNNs), UGAN [33] and FunieGAN [34], based on generative adversarial networks (GANs), and U-Trans [35], combining GAN and Transformer. In addition, two advanced models that fuse CNN and Transformer are also included, namely AutoEnhancer [36] and Frequency [37]. By comparing with these representative models, the performance advantages of MixRformer are comprehensively evaluated.

4.2.1. Qualitative Evaluation

Figure 7 is the result obtained through qualitative evaluation. Specifically, each model enhances underwater images to varying degrees, but the results are unsatisfactory.

Figure 8 shows the results of qualitative evaluation on the UFO-120 dataset to demonstrate generalization ability. It can be seen that each model has different degrees of generalization to enhance underwater images, but the effect is not ideal.

It can be seen that BRUE, based on non-physical models, cannot specifically restore distorted underwater images, and the restored images show exposure and color distortion. UDCP and IBLA are based on physical models; although the former can improve the brightness of the red channel to a certain extent, it cannot restore distorted underwater images, the background of the image deepens the blue-green tone, and the image details and textures are greatly lost. Due to the limitations of light propagation characteristics, the latter still shows a blue-green bias after enhancement, and the local contrast is over-enhanced. ShallowUWnet and Uice2Net are models based on CNN architecture. Compared with previous models, it is obvious that they have improved in effectively correcting local color deviation and enhancing texture details. However, there is an overall lack of brightness, the imbalance of color control in local areas leads to overexposure, and the plants in the picture are not green, but enhanced in yellow, which eventually leads to color deviation. After the emergence of GAN networks, UGAN and FunieGAN, the background image is no longer dark overall, and the enhancement effect is gradually closer to the real image. From the effects shown in Figures 1 and 5, it can be found that due to the severe scattering in the underwater environment, the image generated by UGAN is yellowish, the highlight part has overexposure artifacts, and the edge is distorted; FunieGAN, as can be clearly seen from Figures 3 and 5, can significantly improve the edge clarity of the image. The color contrast of the background in the figure is most similar to our Mixformer. However, like UGAN, the color of the seaweed in Figure 1 is still yellow. The U-Trans network has unstable training factors. Compared with the previous model, Figure 1 significantly restores the green color of the seaweed, but the hazy part in Figure 4 makes the enhanced image blurry. The AutoEnhancer and Frequency algorithms that aggregate CNN and Transformer perform well in visual effects. The images enhanced by visual perception are excellent in high-frequency texture sharpness and low-frequency color fidelity, which are comparable to this method, but the former still has excessive yellowing in some areas in detail texture restoration.

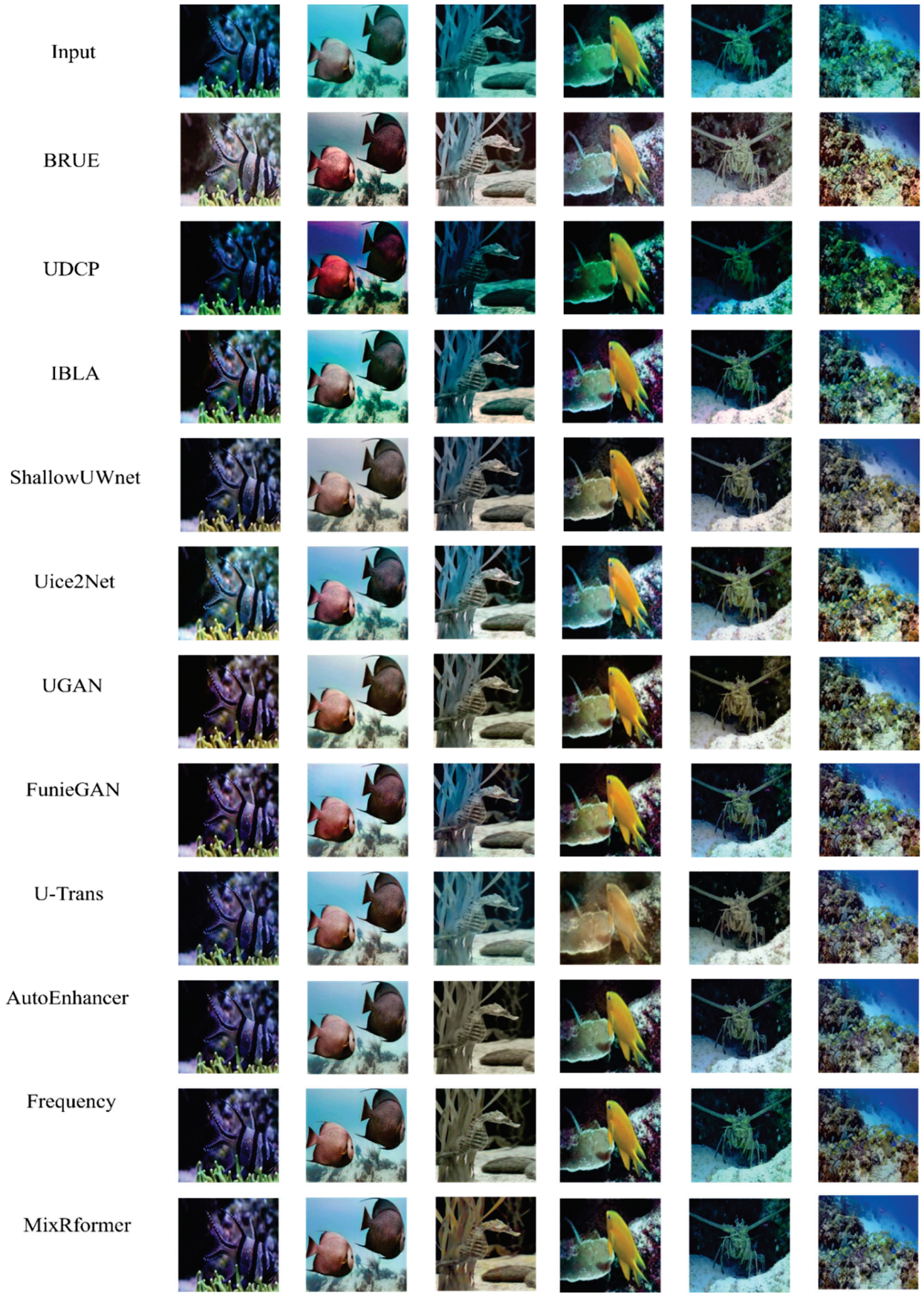


Figure 7. Qualitative comparison results of MixRformer and other methods on EUVP.

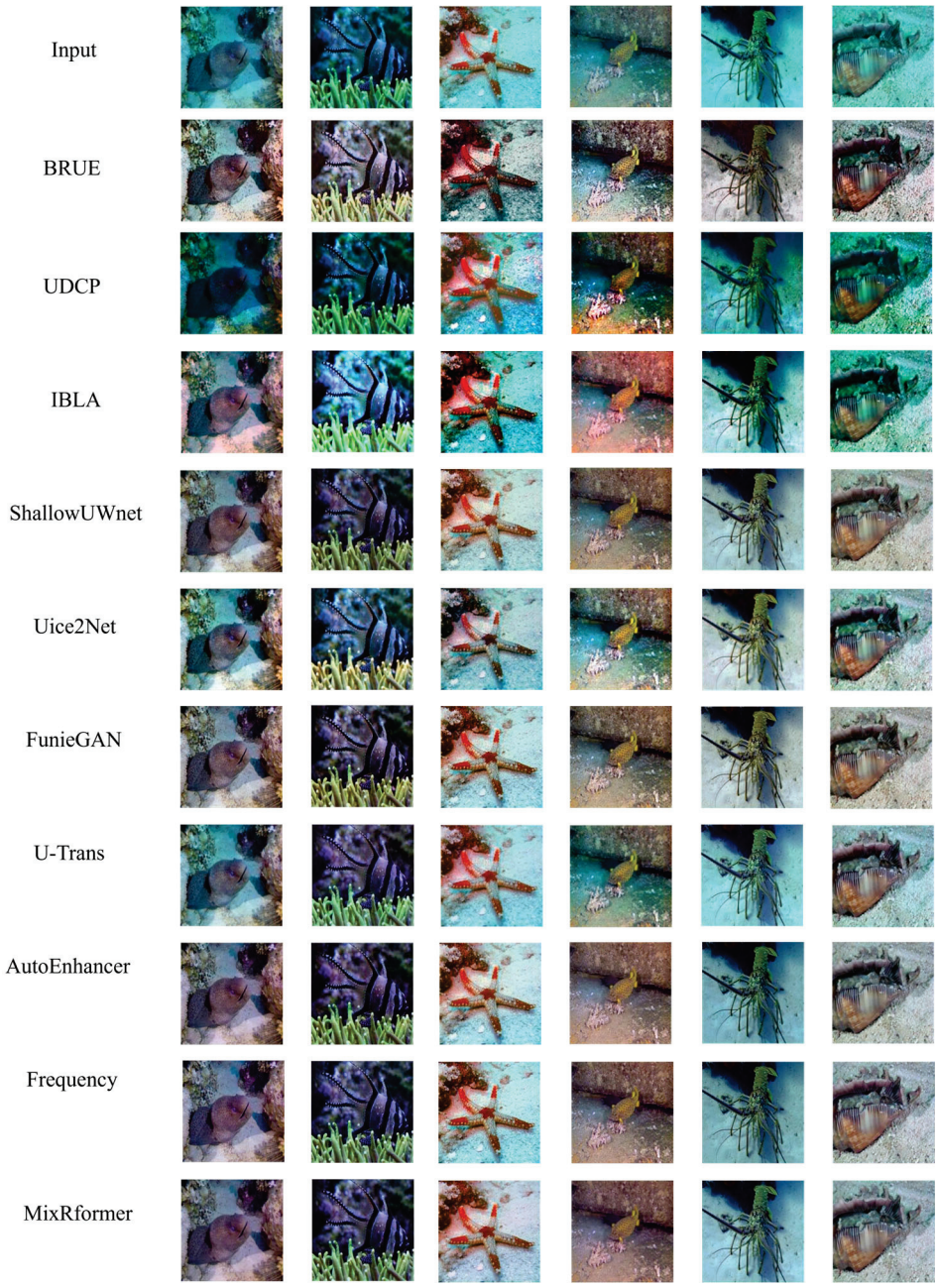


Figure 8. Qualitative comparison results of MixRformer and other methods on UFO-120.

In addition, we use the UFO-120 test set to qualitatively verify the generalization ability of this model. BRUE has high color saturation and contrast, which can effectively enhance the visual impact of the image, but the color is easy to deviate from the sense of reality. UDCP performs well in enhancing the blue-green tone and restores the details well, but some images have slight over-sharpening. IBLA is still the algorithm with the worst processing effect and fails to effectively remove the blue-green bias. Shallow-UWnet, Uice2Net, FunieGAN, and U-Trans still cannot effectively solve the color bias problem. The images processed by the first three models are obviously yellowish, and U-Trans is no longer yellowish, but the excessive increase of green tones makes the image too bright; however, they are worthy of recognition in restoring image details. AutoEnhancer, Frequency and our model, MixRformer, can effectively improve image quality and restore image color, but AutoEnhancer has deficiencies in processing image brightness, and Frequency has certain color distortion problems, but our model achieves very similar results.

In contrast, we can see that by using our Mixformer to obtain low-frequency and high-frequency information through wavelet transformation and dual-branch feature extraction module, the enhanced image is not only closer to the reference image in terms of detail texture and color restoration, but also has a more moderate contrast and more natural color performance.

4.2.2. Quantitative Evaluation

In addition to the qualitative evaluation, we also conducted a comprehensive evaluation of the performance of the model on the EUVP dataset in the quantitative evaluation. It can be clearly seen from Table 1 that our MixRformer achieved the best results in terms of PSNR, SSIM and NIQE evaluation indicators compared with other competing methods, which shows that MixRformer has significant effects in enhancing the detailed texture of underwater images, restoring the structural similarity of images, and enhancing the visual characteristics of images closer to natural images. From the various indicators obtained, it can be seen that the underwater images enhanced by MixRformer not only have a good improvement in contrast, but the model also has a good correction effect on the color shift of underwater images relative to the reference image. In order to verify the generalization ability of the model, we compared our model and the competitive methods in the test set of the UFO-120 dataset. It can be seen from Table 2 that the results clearly show that our model still achieves the best in terms of PSNR, SSIM and NIQE evaluation indicators, indicating that MixRformer can effectively enhance images in different underwater environments.

Table 1. The following table shows the results of quantitative comparison between our model and various current representative models on EUVP. The bold font represents the result with the highest index.

Model	PSNR	SSIM	UIQM	UCIQE	NIQE
BRUE	17.58	0.67	3.17	0.42	4.92
UDCP	15.84	0.57	2.00	0.52	5.66
IBLA	16.55	0.43	2.25	0.49	5.50
ShallowUWnet	26.75	0.79	2.91	0.398	6.04
Uice2Net	26.83	0.86	2.99	0.396	5.83
UGAN	25.42	0.81	2.94	0.41	5.15
FunieGAN	26.24	0.81	2.89	0.45	5.66
U-Trans	22.57	0.74	3.00	0.38	5.04
AutoEnhancer	28.78	0.86	2.85	0.41	5.06
Frequency	28.81	0.86	2.82	0.42	4.95
Ours	**29.01**	**0.87**	2.88	0.42	**4.916**

PSNR, Peak Signal-to-Noise Ratio; SSIM, Structural Similarity; UIQM, Underwater Image Quality Measurement; UCIQE, Underwater Color Image Quality Evaluation; NIQE, Natural Image Quality Evaluator.

Table 2. The following table shows the results of a quantitative comparison between our model and various current representative models on UFO-120. The bold font represents the result with the highest index.

Model	PSNR	SSIM	UIQM	UCIQE	NIQE
BRUE	17.05	0.64	**3.09**	0.43	4.96
UDCP	19.12	0.66	2.25	**0.51**	5.11
IBLA	20.06	0.71	2.45	0.49	5.06
ShallowUWnet	27.02	0.81	2.89	0.43	5.16
Uice2Net	21.00	0.75	3.02	0.45	5.03
FunieGAN	26.97	0.83	2.73	0.45	4.88

Table 2. *Cont.*

Model	PSNR	SSIM	UIQM	UCIQE	NIQE
U-Trans	25.29	0.80	2.96	0.42	4.99
AutoEnhancer	28.13	0.84	2.79	0.43	4.88
Frequency	28.33	0.84	2.73	0.44	4.91
Ours	**28.43**	**0.85**	2.84	0.43	**4.68**

PSNR, Peak Signal-to-Noise Ratio; SSIM, Structural Similarity; UIQM, Underwater Image Quality Measurement; UCIQE, Underwater Color Image Quality Evaluation; NIQE, Natural Image Quality Evaluator.

4.3. Ablation Study

We use a series of ablation experiments to prove and evaluate the effectiveness of each component in Mixformer.

In our model, the most important ones are WT, ConvBlock and RGLUWin Transformer. In order to verify the effectiveness of each of these blocks, we have performed many ablation experiments to prove it. The WT operation we introduced plays a vital role in shortening the execution time, GPU memory consumption and introducing wavelet information. In order to verify that WT can do these things, we conducted a comparative experiment. From the results in Table 3, it can be seen that when we introduce WT, MixRformer only uses one-fourth of the running time and one-fourth of the GPU memory occupancy to achieve a significant reduction in the amount of calculation and complexity. At the same time, it can be seen in Figure 9 that the overall image without WT is not bright enough and the color is dark. The w/o ConvBlock block basically retains the local features of the image, such as texture, but the contrast of the image is not obvious and the edge information is too blurred. The w/o RGLUWin Transformer block does not effectively eliminate the blue-green tones of the image, and the local area is occupied by blue-green tones. The importance of WT to the model is also clearly seen from the quantitative analysis in Table 4. MixRformer is improved by one point under the effect of WT. In general, compared with our MixRformer which controls the information flow through GLU, ConBlock for surface information, and RGLUWin Transformer for fine-grained extraction, on the basis of reducing resource consumption, the enhanced image not only retains the local detail texture, but also basically eliminates the color deviation in the image, enhancing the visual perception and contrast effect.

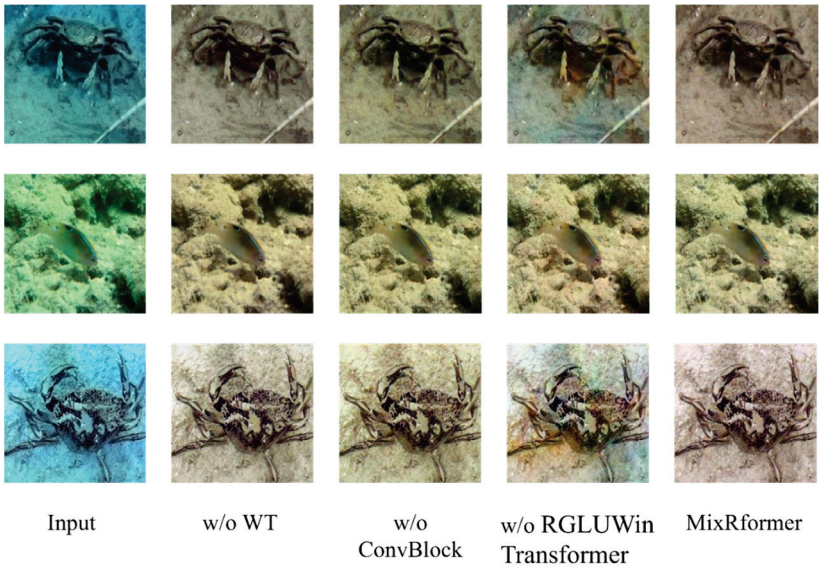


Figure 9. The figure shows the qualitative ablation study of WT, ConvBlock, and RGLUWin Transformer, respectively.

Table 3. Comparison of WT model usage on GPU. The bold font represents the result with the highest index.

Model	Patchsize	GPU	Time/epoch
w/o WT	3	21,976 MiB	66 m
MixRformer	**12**	**23,418 MiB**	**17 m**

Table 4. Quantitative Ablation Study on WT, ConvBlock, and RGLUWin Transformer. The bold font represents the result with the highest index.

Model	PSNR	SSIM	UIQM
w/o WT	28.083 ± 3.369	0.835 ± 0.062	2.859 ± 0.388
w/o ConvBlock	28.521 ± 3.08	0.847 ± 0.054	2.845 ± 0.393
w/o RGLUWin Transformer	28.59 ± 3.163	0.846 ± 0.052	2.88 ± 0.371
MixRformer	**29.01 ± 3.158**	**0.87 ± 0.054**	2.88 ± 0.387

PSNR, Peak Signal-to-Noise Ratio; SSIM, Structural Similarity; UIQM, Underwater Image Quality Measurement.

For the introduced GLU module, we conducted a qualitative and quantitative comparison between GLU and MLP on RWin Transformer. As shown in Figure 10, when using the traditional MLP, the removal of blue-green tones in Figures 1 and 3 is far inferior to the effect of w/GLU. At the same time, from the results of Figure 2, it can be observed that the contrast of marine animals and the sharpness of texture and detail restoration on the animals are all enhanced by GLU better than MLP. At the same time, from the quantitative aspect, it can be found from Table 5 that GLU is stronger than MLP in both information suppression and selection.

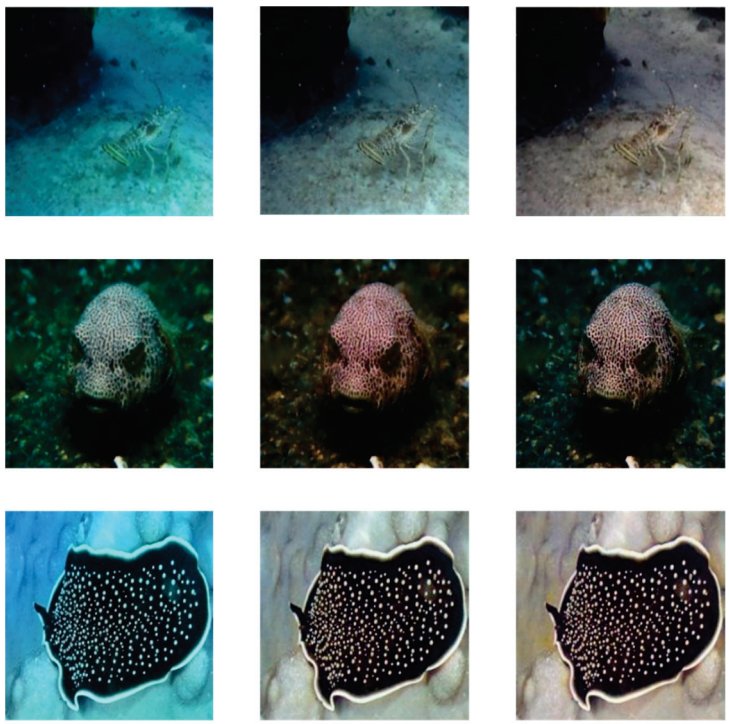


Figure 10. Qualitative comparison between MLP and GLU.

Table 5. The following table shows a quantitative comparison of MLP and GLU. The bold font represents the result with the highest index.

Model	PSNR	SSIM	UIQM
w/MLP	28.89 ± 3.222	0.854 ± 0.053	2.873 ± 0.402
MixRformer w/GLU	**29.01 ± 3.158**	**0.87 ± 0.054**	**2.88 ± 0.387**

PSNR, Peak Signal-to-Noise Ratio; SSIM, Structural Similarity; UIQM, Underwater Image Quality Measurement.

Of course, in order to verify the effectiveness of the components of the multi-loss function we set in achieving high-quality image restoration, we conducted an ablation experiment on the loss function. It can be clearly seen from Figure 11 that when w/o VGG, the image restoration is also relatively clear, and the texture details are gradually restored. However, the image is obviously reddish, and the contrast and generalization ability are reduced, and the color calibration is offset, as in Figures 1 and 2. In contrast, using a complete multi-loss function helps to provide more comprehensive and fine texture details and a contrast closer to the reference image. It can be clearly seen from the data in Table 6 that when using a multi-loss function, the PSNR and SSIM indicators are significantly improved in value, so that the network can restore and enhance the image from multiple angles, and achieve enhanced generalization ability for complex underwater environments.

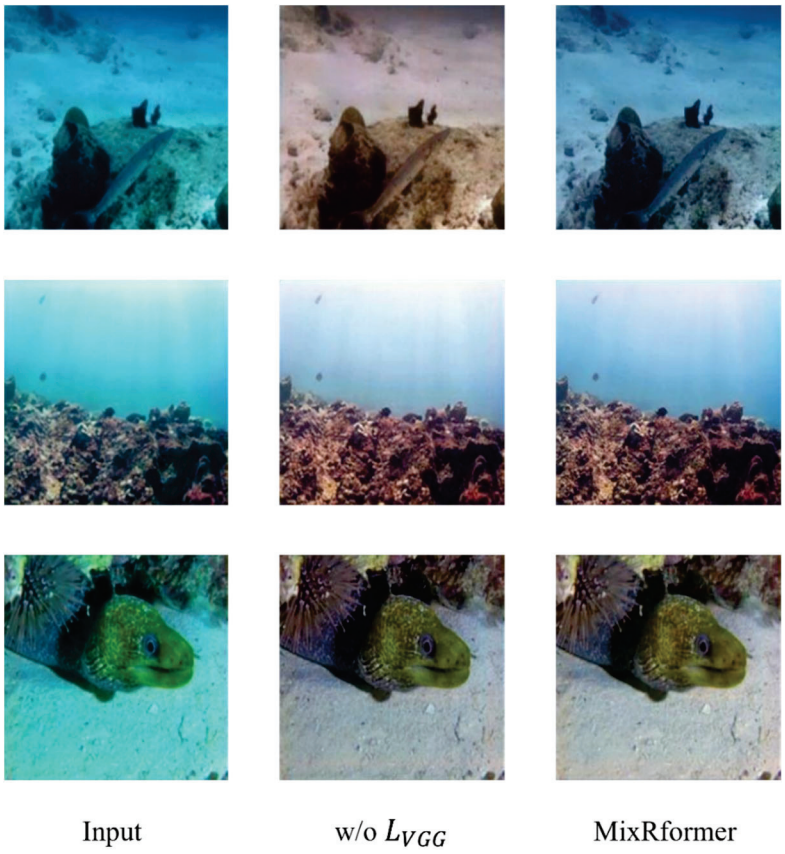


Figure 11. Qualitative ablation experiments on each of the multiple loss functions.

At the same time, in order to verify the effectiveness of wavelet transformation on the number of parameters and FLOPs of our method, we compared the number of parameters occupied by different modules in our method. The specific results are shown in Table 7. It can be clearly seen that when we introduce wavelet transformation, without basically increasing the number of model parameters, it not only greatly reduces the FLOPs of the model, but also improves the visualization effect of the image and improves the quantitative

evaluation index. This shows that MixRformer, while reducing FLOPs, obtains satisfactory visual effects and quantitative indicators.

Table 6. The following table is a quantitative ablation study of the loss function.

Model	PSNR	SSIM	UIQM
w/o L_{VGG}	28.669 ± 3.136	0.85 ± 0.052	2.841 ± 0.412
MixRformer w/GLU	**29.01 ± 3.158**	**0.87 ± 0.054**	**2.88 ± 0.387**

PSNR, Peak Signal-to-Noise Ratio; SSIM, Structural Similarity; UIQM, Underwater Image Quality Measurement.

Table 7. The following table is a quantitative comparison of the number of parameters and FLOPs.

Model	Parameters	FLOPs
w/o ConvBlock	10.10 M	11.58 G
w/o RGLUWin	12.97 M	15.29 G
Transformer	20.05 M	100.16 G
w/o WT		
MixRformer	**20.06 M**	**25.17 G**

5. Conclusions

In this paper, we propose the MixRformer model, a novel method for underwater image enhancement. In detail, we introduce the wavelet transform into our model, which uses low-frequency information for image enhancement, restores the texture and detail information of the image, and greatly reduces the image resolution, reduces GPU memory consumption, and reduces computational cost. At the same time, this paper proposes a dual-branch feature extraction module (DFCB) that combines the advantages of CNN and Transformer to fully restore image details by extracting local and global features. In addition, we also design multiple loss functions based on the traditional loss function to meet multiple constraints and realize the model's adaptability to complex environments. Extensive ablation experiments show that the important components of our MixRformer improve the model performance both quantitatively and qualitatively, and the model execution time and GPU memory consumption have been significantly improved. In the future, we will better optimize MixRformer for tasks related to underwater video enhancement.

Author Contributions: Conceptualization, J.L. and L.Z.; methodology, J.L.; software, J.L.; validation, J.L., H.L. (Heng Li) and X.X.; formal analysis, J.L.; investigation, J.L. and H.L. (Heng Li); resources, H.L. (Hui Liu); data curation, L.Z.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and H.L. (Heng Li); visualization, J.L.; supervision, J.L. and H.L. (Heng Li); project administration, X.X.; funding acquisition, H.L. (Hui Liu). All authors have read and agreed to the published version of the manuscript.

Funding: Yunnan Provincial Science and Technology Department Basic Research Project-General-Project, 202401AT070375, and Yunnan Provincial University Service Key Industry Science and Technology Program-General Project, FWCYQYCT2024003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are from public datasets, which are introduced in Section 4.1.

Acknowledgments: We extend our sincere appreciation to the funders of this project and all the 509-510 teams and individuals who contributed to this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 2341–2353. [PubMed]
2. Drews, P.; Nascimento, E.; Moraes, F.; Botelho, S.; Campos, M. Transmission estimation in underwater single images. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 825–830.
3. Zhang, S.; Wang, T.; Dong, J.; Yu, H. Underwater image enhancement via extended multi-scale Retinex. *Neurocomputing* **2017**, *245*, 1–9. [CrossRef]
4. Drews, P.L.J.; Nascimento, E.R.; Botelho, S.S.C.; Campos, M.F.M. Underwater depth estimation and image restoration based on single images. *IEEE Comput. Graph. Appl.* **2016**, *36*, 24–35. [CrossRef]
5. Chen, X.; Zhang, P.; Quan, L.; Yi, C.; Lu, C. Underwater image enhancement based on deep learning and image formation model. *arXiv* **2021**, arXiv:2101.00991.
6. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef] [PubMed]
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
8. Zhang, D. Wavelet transform. In *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*; Springer International Publishing: Cham, Switzerland, 2019; pp. 35–44.
9. Chen, Z.; Zhang, Y.; Gu, J.; Kong, L.; Yuan, X. Cross aggregation transformer for image restoration. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 25478–25490.
10. Hou, G.; Pan, Z.; Wang, G.; Yang, H.; Duan, J. An efficient nonlocal variational method with application to underwater image restoration. *Neurocomputing* **2019**, *369*, 106–121. [CrossRef]
11. Zhou, Y.; Wu, Q.; Yan, K.; Feng, L.; Xiang, W. Underwater image restoration using color-line model. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 907–911. [CrossRef]
12. Song, W.; Wang, Y.; Huang, D.; Liotta, A.; Perra, C. Enhancement of underwater images with statistical model of background light and optimization of transmission map. *IEEE Trans. Broadcast.* **2020**, *66*, 153–169. [CrossRef]
13. Hou, G.; Li, J.; Wang, G.; Yang, H.; Huang, B.; Pan, Z. A novel dark channel prior guided variational framework for underwater image restoration. *J. Vis. Commun. Image Represent.* **2020**, *66*, 102732. [CrossRef]
14. Li, C.; Anwar, S.; Porikli, F. Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognit.* **2020**, *98*, 107038. [CrossRef]
15. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
16. Haiyang, Y.; Ruige, G.; Zhongda, Z.; Yuzhang, Z.; Xiaobo, Z.; Tao, L.; Haiyan, W. U-TransCNN: A U-shape transformer-CNN fusion model for underwater image enhancement. *Displays* **2025**, *88*, 103047. [CrossRef]
17. Chen, H.Q.; Shen, X.; Zhao, Z.; Yan, Y. Hybrid CNN-Transformer Network for Two-Stage Underwater Image Enhancement with Contrastive Learning. In Proceedings of the 2024 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Bali, Indonesia, 19–22 August 2024; pp. 1–5.
18. Wang, Y.; Hu, S.; Yin, S.; Deng, Z.; Yang, Y.-H. A multi-level wavelet-based underwater image enhancement network with color compensation prior. *Expert Syst. Appl.* **2024**, *242*, 122710. [CrossRef]
19. Zhao, C.; Cai, W.; Dong, C.; Hu, C. Wavelet-based fourier information interaction with frequency diffusion adjustment for underwater image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 8281–8291.
20. Liu, S.; Fan, H.; Wang, Q.; Han, Z.; Guan, Y.; Tang, Y. Wavelet–pixel domain progressive fusion network for underwater image enhancement. *Knowl.-Based Syst.* **2024**, *299*, 112049. [CrossRef]
21. Wang, Z.; Tao, H.; Zhou, H.; Deng, Y.; Zhou, P. A content-style control network with style contrastive learning for underwater image enhancement. *Multimed. Syst.* **2025**, *31*, 1–13. [CrossRef]
22. Chen, X.; Tao, H.; Zhou, H.; Zhou, P.; Deng, Y. Hierarchical and progressive learning with key point sensitive loss for sonar image classification. *Multimed. Syst.* **2024**, *30*, 1–16. [CrossRef]
23. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2022; pp. 205–218.
24. Chen, Y.; Zhang, X.; Peng, L.; He, Y.; Sun, F.; Sun, H. Medical image segmentation network based on multi-scale frequency domain filter. *Neural Netw.* **2024**, *175*, 106280. [CrossRef] [PubMed]
25. Shi, D. Transnext: Robust foveal visual perception for vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 17773–17783.

26. Yang, Q.; Yan, P.; Zhang, Y.; Yu, H.; Shi, Y.; Mou, X.; Kalra, M.K.; Zhang, Y.; Sun, L.; Wang, G. Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE Trans. Med. Imaging* **2018**, *37*, 1348–1357. [CrossRef]
27. Shin, Y.S.; Cho, Y.; Pandey, G.; Kim, A. Estimation of ambient light and transmission map with common convolutional architecture. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–7.
28. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
29. Zhuang, P.; Li, C.; Wu, J. Bayesian retinex underwater image enhancement. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104171. [CrossRef]
30. Peng, Y.T.; Cosman, P.C. Underwater image restoration based on image blurriness and light absorption. *IEEE Trans. Image Process.* **2017**, *26*, 1579–1594. [CrossRef]
31. Naik, A.; Swarnakar, A.; Mittal, K. Shallow-uwnet: Compressed model for underwater image enhancement (student abstract). In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 19–21 May 2021; Volume 35, pp. 15853–15854.
32. Wang, Y.; Guo, J.; Gao, H.; Yue, H. UIEC^ 2-Net: CNN-based underwater image enhancement using two color space. *Signal Process. Image Commun.* **2021**, *96*, 116250. [CrossRef]
33. Fabbri, C.; Islam, M.J.; Sattar, J. Enhancing underwater imagery using generative adversarial networks. In Proceedings of the 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7159–7165.
34. Islam, M.J.; Xia, Y.; Sattar, J. Fast underwater image enhancement for improved visual perception. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3227–3234. [CrossRef]
35. Peng, L.; Zhu, C.; Bian, L. U-shape transformer for underwater image enhancement. *IEEE Trans. Image Process.* **2023**, *32*, 3066–3079. [CrossRef] [PubMed]
36. Tang, Y.; Iwaguchi, T.; Kawasaki, H.; Sagawa, R.; Furukawa, R. AutoEnhancer: Transformer on U-Net architecture search for underwater image enhancement. In Proceedings of the Asian Conference on Computer Vision, Macao, China, 4–8 December 2022; pp. 1403–1420.
37. Wang, D.; Sun, Z. Frequency domain based learning with transformer for underwater image restoration. In *Pacific Rim International Conference on Artificial Intelligence*; Springer Nature: Cham, Switzerland, 2022; pp. 218–232.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

High-Speed Multiple Object Tracking Based on Fusion of Intelligent and Real-Time Image Processing

Yuki Kawawaki [1,*,†] and Yuji Yamakawa [2]

[1] Graduate School of Engineering, The University of Tokyo, Tokyo 153-8505, Japan
[2] Interfaculty Initiative in Information Studies, The University of Tokyo, Tokyo 153-8505, Japan; y-ymkw@iis.u-tokyo.ac.jp
* Correspondence: kawawaki-yuki628@g.ecc.u-tokyo.ac.jp
† Current address: Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan.

Abstract: Multiple object tracking (MOT) is a critical and active research topic in computer vision, serving as a fundamental technique across various application domains such as human–robot interaction, autonomous driving, and surveillance. MOT typically consists of two key components: detection, which produces bounding boxes around objects, and association, which links current detections to existing tracks. Two main approaches have been proposed: one-shot and two-shot methods. While previous works have improved MOT systems in terms of both speed and accuracy, most works have focused primarily on enhancing association performance, often overlooking the impact of accelerating detection. Thus, we propose a high-speed MOT system that balances real-time performance, tracking accuracy, and robustness across diverse environments. Our system comprises two main components: (1) a hybrid tracking framework that integrates low-frequency deep learning-based detection with classical high-speed tracking, and (2) a detection label-based tracker management strategy. We evaluated our system in six scenarios using a high-speed camera and compared its performance against seven state-of-the-art (SOTA) two-shot MOT methods. Our system achieved up to 470 fps when tracking two objects, 243 fps with three objects, and 178 fps with four objects. In terms of tracking accuracy, our system achieved the highest MOTA, IDF1, and HOTA scores with high-accuracy detection. Even with low detection accuracy, it demonstrated the potential of long-term association for high-speed tracking, achieving comparable or better IDF1 scores. We hope that our multi-processing architecture contributes to the advancement of MOT research and serves as a practical and efficient baseline for systems involving multiple asynchronous modules.

Keywords: multiple object tracking; high-speed processing; deep learning; hybrid tracking; multi-processing; tracker management

1. Introduction

Multiple object tracking (MOT) is a crucial and active research topic in computer vision, serving as a fundamental technique across various application domains. These include human–robot interaction [1], autonomous driving [2], surveillance [3], sports video analysis [4], and medical imaging [5]. Robots are increasingly being integrated into human environments, creating a growing demand for systems that enable them to operate effectively in diverse settings and adapt seamlessly to dynamic surroundings. To enhance both robot manipulability and human safety, continuous, accurate, and real-time tracking is essential for downstream tasks like motion planning, reasoning, and

collision avoidance [6,7]. Similarly, in autonomous driving, it is critical to detect and track pedestrians emerging from blind spots as early as possible to ensure safety [8]. Thus, an MOT system that balances real-time performance and tracking accuracy is vital for the advancement of these downstream technologies.

MOT is typically approached in two stages: detection, which produces a set of bounding boxes around objects, and association, which links current detections to existing tracks. There are two approaches: one-shot and two-shot methods. One-shot MOT methods unify detection and re-identification (re-ID) within a single network. Previously, it suffered from slow processing time, but FairMOT achieved relatively fast speeds (30 fps) while maintaining tracking accuracy by balancing detection and re-ID tasks [9]. OffsetNet further improved tracking performance and processing speed (50 fps) by integrating amodal bounding box detection, instance segmentation, and tracking into a unified framework [10]. While these one-shot methods achieve strong tracking performance, they still generally fall short of the processing speed offered by geometry-based two-shot approaches.

For online methods, two-shot approaches, tracking-by-detection (TBD), have demonstrated state-of-the-art performance. These methods leverage deep learning-based detectors and employ the Hungarian algorithm [11] for effective inter-frame track assignment. Common association metrics include intersection over union (IoU), Mahalanobis distance, and appearance similarity (e.g., cosine similarity between embedding vectors). To enable real-time processing, lightweight association methods rely primarily on motion and bounding box data. Kalman Filter-based prediction has been widely adopted to track objects in dynamic scenes [12]. However, simple Kalman Filters often struggle with tracker ID switching and are not well suited for long-term tracking. ByteTrack improved association by introducing a two-stage matching strategy [13], while Observation-Centric SORT (OC-SORT) [14] enhanced SORT performance by reducing Kalman Filter error accumulation through virtual trajectory estimation and direction-aware noise modeling. These methods offer a favorable trade-off between accuracy and processing speed among conventional methods. In contrast, some approaches incorporate deep learning-based appearance features to achieve more robust long-term tracking under occlusion, though often at the cost of slower inference speeds [15,16].

Thus, conventional research has advanced MOT systems in terms of processing speed and tracking accuracy. However, current state-of-the-art (SOTA) online methods, such as ByteTrack and OC-SORT, rely on deep learning-based detectors, where the inference time becomes a bottleneck for further speed improvements. To address this, we propose a high-speed MOT system that balances real-time performance, tracking accuracy, and environment-invariant versatility. Our system is composed of two main components: a hybrid tracking framework and a detection label-based tracker management strategy. The hybrid tracking framework integrates three asynchronous modules—deep learning-based detection, classical high-speed detection methods (correlation filters and template matching), and Kalman prediction—to enable high-speed processing and mutually supportive long-term tracking. The detection label-based tracker management synchronously handles multiple trackers across asynchronous threads while maintaining low latency. To further enhance robustness and continuity, we introduce three techniques: dynamic search area adjustment, duplicate tracker elimination, and update skipping for occlusion-aware templates. We evaluate our system in six scenarios using a high-speed camera and compare its performance against current two-shot SOTA methods. The demonstration video is available in [17].

The main contributions of this paper are:

- A holistic multiprocessing algorithm architecture running multiple modules that mutually assist one another.

- High-speed MOT system for multiple categories of objects.
- Efficient detection label-based management of trackers which synchronously handles trackers from asynchronous modules.
- Extensive evaluations of MOT scenarios with a high-speed camera, which have not been worked on.

This paper is organized as follows. Section 2 reviews related work in tracking systems. Section 3 outlines the proposed methods, and effectiveness is demonstrated in Section 4. In Section 5, we evaluate an effect of the individual proposed method. Finally, Section 6 summarizes the proposed method and discusses potential future work.

2. Related Work

This section explores two kinds of previous works about tracking methods and Multiple Object Tracking (MOT). Afterward, we provide insights into the strengths and limitations of existing approaches, and we illustrate the novelty of our proposed method.

2.1. Tracking Method

We give comprehensive reviews about tracking methods of template image-based, motion-based, deep learning and hybrid methods. For template image-based tracking, template matching [18], calculating the similarity between a pre-defined template image and the current frame, is widely used, but it is not adapted to rotational and scale changes. Additionally, correlation filter-based methods are known for their balancing robustness to scale and rotational changes and processing speed [19–25]. They employ the fast Fourier transform (FFT) to reduce the computational load required to create and update template images. In particular, the Minimum Output Sum of Squared Error (MOSSE) tracker utilizes raw images to calculate correlation filters, achieving high-speed processing rates of over 500 fps [20]. However, this method is sensitive to changes in appearance, making it less suitable for long-term tracking. While other derivative methods offer higher accuracy, they often sacrifice processing speed. For motion-based tracking, calculating optical flow, which represents the distribution of apparent velocity, is widely used. There are two main methods: feature-based sparse optical flow and pixel-wise dense optical flow [26–28]. In sparse optical flow, only selected features are tracked, allowing for high-speed processing. However, finding appropriate and robust features can be challenging. In contrast, some dense optical flow methods, which analyze every pixel, have achieved high-speed processing rates of over 300 fps [27,28]. While it is useful to extract motion of targets with vague boundaries like a human's joint, it is usually sensitive to background noise. Deep learning-based methods are also employed for versatile tracking applications. For instance, Generic Object Tracking Using Regression Networks (GOTURN), which relies on offline training, achieved tracking of unknown objects at 100 fps [29]. However, GOTURN is sensitive to changes in appearance. Siamese Network-based tracking methods have been widely developed, offering promising results [30–34]. For accurate tracking, the Multi-Domain Convolutional Neural Network (MDNet) combines both offline and online training, but its high computational requirements result in slow processing speed [35]. To achieve a high-speed and robust tracking system, hybrid tracking methods has been tackled. Nishimura et al. [36] achieved robust tracking by integrating several correlation filter-based methods. In [37–40], a tracking method has been developed that combines deep learning-based detection with template-based tracking, incorporating a retracking mechanism in case the tracking score is low. The work by Matsuo et al. [41] succeeded in tracking a transparent object with low latency (618 fps) combining deep learning-based detection [42], motion detection, and correlation filter-based tracking [20]. Our proposed method was inspired by this research. However, these methods typically focus on a single

target, so consistent, efficient management of asynchronous information is required for more complex scenarios such as MOT.

2.2. Multiple Object Tracking (MOT)

In general tracking scenarios, the problem of multiple object tracking (MOT) is typically addressed, which involves both detection and association tasks. There are two main types of models: one-shot and two-shot methods.

One-shot methods have addressed challenges such as task conflict between detection and re-identification (re-ID), occlusion handling, and real-time performance. FairMOT [9] achieved 30 fps and improved tracking accuracy by balancing detection and re-ID, extracting features at object centers to avoid the misalignment issue seen in the Joint Detection and Embedding (JDE) model [43]. Occlusion remains a major challenge in MOT. To address this, Trackformer [44] introduced attention mechanisms to enhance tracking robustness, while Preserved ID MOT (PID-MOT) [45] detects appearance changes by predicting object visibility levels and refining features using a visibility-guided association strategy. Furthermore, CSMOT [46] improved re-ID discriminability through angle-center loss and coordinate attention, enhancing small object detection. In terms of real-time performance, OffsetNet (50 fps) [10] unified detection, segmentation, and tracking using pixel-offset representations. Despite these advancements, one-shot methods often remain slower than geometry-based two-shot approaches.

Among two-shot methods, Simple Online and Realtime Tracking (SORT) [12] is a foundational and widely used approach. SORT tracks multiple objects by combining deep learning-based detection with a position-based tracking system using the Kalman Filter and Hungarian algorithm. Although effective, it struggles with irregular motion and frequent identity switches in crowded scenes. Several derivative methods have been proposed. ByteTrack [13] introduced a two-stage matching scheme; BoT-SORT [47] incorporated a revised Kalman Filter and camera motion compensation; OC-SORT [14] carefully reduced Kalman Filter error accumulation. Other methods integrated deep learning-based appearance features, such as DeepSORT [16] and StrongSORT [15], the latter addressing missing detections and associations via the Appearance-Free Link (AFLink) model and Gaussian-Smoothed Interpolation (GSI). BoT-SORT-ReID [47] and Deep OC-SORT [48] have also been proposed. BoostTrack++ [49] is based on BoostTrack [50] by boosting confidence for likely detections and introducing Soft Buffered IoU (BIoU), a similarity measure that combines shape, Mahalanobis distance, and a novel soft IoU metric to refine detection–tracklet association. Improved Association (ImprAssoc) [51] enhanced association performance through a combined matching scheme, integrating distance metrics and occlusion-aware initialization. In summary, two-shot methods have made significant strides in improving both tracking accuracy and processing speed, though much of the focus has been on association techniques rather than accelerating the detection process.

2.3. Challenges of Previous Studies

Previous hybrid approaches primarily focused on simple scenarios, such as single-target tracking. However, in more complex contexts like multiple object tracking (MOT), consistent and efficient management of asynchronous information is essential. In contrast, conventional two-shot MOT systems have mainly prioritized the association process, often placing less emphasis on increasing detection inference speed. As a result, as shown in Table 1, existing systems struggle to meet all key requirements—low latency, high tracking quality, and robustness across diverse environments and target types. To address this, our research aims to develop a versatile tracking system that fulfills all these criteria by

accelerating detection and tracker detection label-based tracker management as shown in Figure 1.

Table 1. Comparison between conventional methods and the proposed method. ✓represents "good quality".

Method	Low Latency	Accuracy	Environments	Types and Number of Targets
Hybrid method [41]	✓	✓	✓	
One-shot method [9,10]		✓	✓	✓
Two-shot method [13,14]		✓	✓	✓
Proposed Method	✓	✓	✓	✓

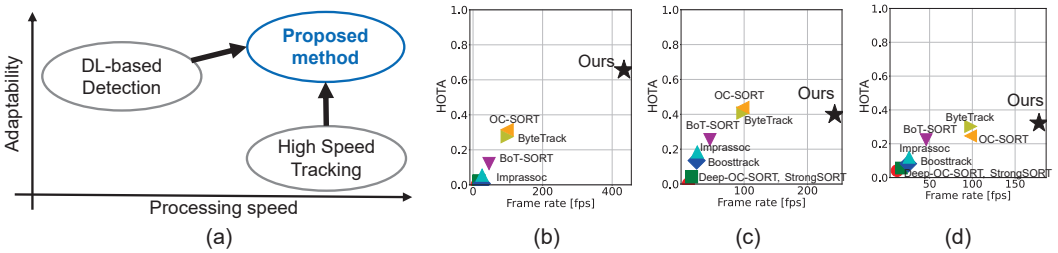


Figure 1. Our motivation and achievements. (**a**) Our motivation. (**b**–**d**) Value of HOTA metric and frame rate for various two-shot tracking methods with (**b**) 2-ball (Section 5.2), (**c**) 3-ball (Section 4.3.4) and (**d**) 4-ball (Section 4.3.5) scenarios.

3. Proposed Methodology

3.1. Overview

In Figure 2, we compare the proposed architecture with a conventional two-shot MOT system [12]. The conventional system uses geometric predictions to match trackers, processing the entire pipeline sequentially. In contrast, the proposed system runs three threads—detection, tracking, and database—in parallel. This parallel processing allows for faster and more accurate tracking by utilizing not only the predicted values but also high-speed actual tracking data. Additionally, the proposed system detect objects using a deep learning-based method in the detection thread, making our system applicable in a wide range of environments without the need for markers, while maintaining processing speed. This architecture can be extended to systems that aim for efficiency and precision by integrating multiple methods.

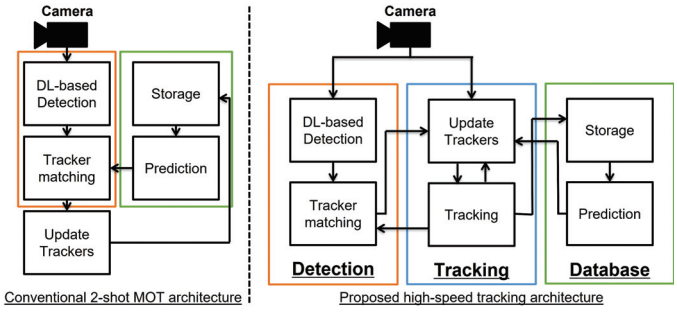


Figure 2. Tracking architecture of the conventional two-shot multiple object tracking and the proposed high-speed tracking systems. The orange box represents the detection module, the blue box denotes the tracking module, and the green box illustrates the database module.

Figure 3 illustrates a detailed flowchart of the proposed system. Our approach integrates deep learning-based low-frequency detection, classical high-frequency detection methods—MOSSE and template matching—and Kalman Filter prediction, allowing these components to support each other and achieve robust tracking performance. In the tracking

thread, we employ the correlation filter-based method, MOSSE, to handle changes in the target object's scale and rotation, while template matching offers additional support. Both methods operate concurrently, and their effectiveness is discussed in Section 5.

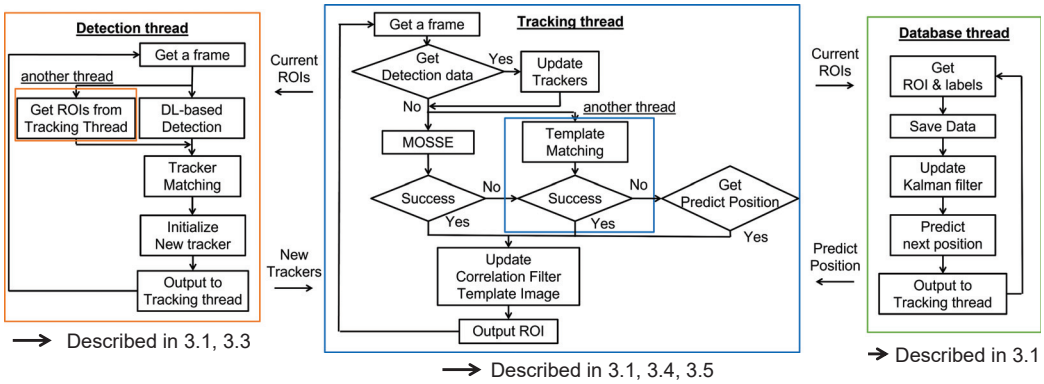


Figure 3. Flowchart of the proposed multiple object tracking system. The orange box represents the detection thread, the blue box represents the tracking thread, and the green box represents the database thread.

The tracker's template image in the tracking thread is updated using results from both the detection thread and the previous step of the tracking thread. For the tracker's position, we use either the tracking thread or the database thread—based on Kalman Filter prediction, which offers faster updates than deep learning-based detection. In the database thread, Kalman Filter models with constant velocity assumptions are updated using results from the tracking thread. Further details on data exchange and tracker management are provided in Section 3.2.

3.2. Detection Label-Based Tracker Management

We describe how multiple trackers are managed based on detection labels.

First, we introduce the structure of each tracker and the types of data exchanged between threads. The state of each tracker is modeled as $\{label, x, y, \dot{x}, \dot{y}, w, h, CF, template, trackingScore\}$. Here, *label* is the detection label; x and y represent the center position of the ROI; $\dot{x}$ and $\dot{y}$ represent the velocity; w and h denote the width and height of the ROI; *CF* represents the correlation filter used in MOSSE; *template* refers to the template image; and *trackingScore* indicates the score from each tracking method, including the Peak-to-Sidelobe Ratio (PSR) and Zero-mean Normalized Cross-Correlation (ZNCC).

Next, we explain the communication among the detection, tracking, and database threads. Exchanges between the detection and tracking threads include template images and their corresponding bounding box information. Details of this communication are explained in Sections 3.3 and 3.4. Concerning exchanges between tracking and databased threads, the tracking thread sends updated tracker information—such as the latest geometric data and a list of trackers marked for removal—to the database thread, which uses this data to update its Kalman Filter models. When each Kalman Filter is sufficiently updated (i.e., the update count reaches $\text{Counter}_{\text{valid}}$), the database thread returns predicted positions to the tracking thread.

We illustrate the detection label-based tracker management process with a simple example in Figure 4. The indexing rule is straightforward; non-negative indices (e.g., '0' and '1') represent valid detection labels, while lost trackers are indexed as '−1'. The tracker container preserves the order of trackers throughout the system to ensure consistent and synchronous tracking.

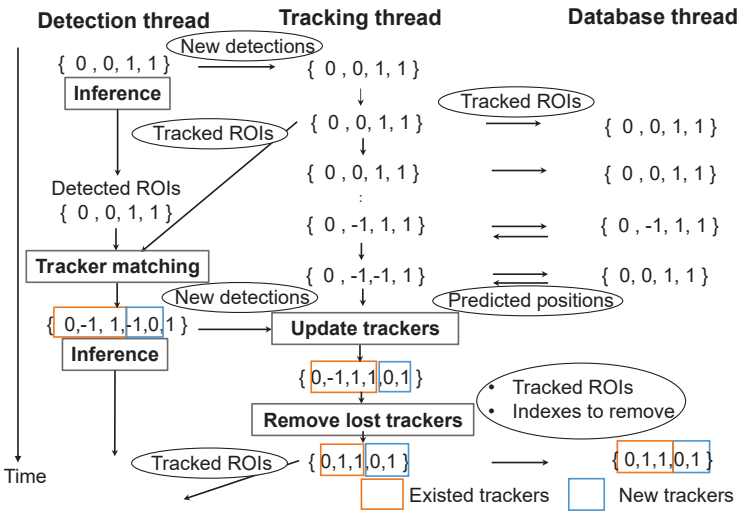


Figure 4. Detection label-based tracker management. $\{0, 0, 1, 1\}$ represents a list of detection labels. The arrows indicate data flow among the detection, tracking, and database threads. Rectangular boxes denote processes, while circular nodes represent exchanged data. Orange boxes indicate existing trackers, and blue boxes indicate new trackers detected by the detection thread.

We now explain how the tracker container is synchronized and updated. To synchronize the next tracker candidates between the detection and tracking threads, as soon as the tracking thread updates tracker information based on detection results, it sends the latest tracker data to the detection thread. This information is first passed through a buffer thread to avoid interfering with deep learning inference.

Tracker removal is handled carefully to maintain consistency across threads. If a tracker has not been updated for several frames (i.e., exceeds $Counter_{lost}$), it is marked for removal. Since the number of trackers should only change when new deep learning-based detection results are available, removal occurs only when the tracking thread receives new tracker data from the detection thread. Trackers marked for deletion are removed when the tracking thread sends the latest tracker list to the detection and database threads, as illustrated in the "Remove lost trackers" part of Figure 4. Trackers with label '$-$' in the orange box are removed from both the tracking and detection threads in this example. As mentioned above, to keep the database thread synchronized, the tracking thread sends not only the updated ROI information but also the indices of the removed trackers.

This detection label-based management strategy enables our MOT system to track objects efficiently with low latency, regardless of object type or quantity.

3.3. Tracker Matching

The matching process between trackers in the detection and tracking threads is illustrated in Figure 5, which shows sequential data and the tracking state at the beginning of deep learning-based detection inference. As depicted, the tracking thread operates with low latency, resulting in minimal deviation between the actual and tracked positions. Therefore, by using the most recent tracked ROIs available in the tracking thread, the proposed system can accurately associate trackers between the detection and tracking threads.

To perform tracker matching, we apply the Hungarian algorithm [11], with the cost defined as follows in Equation (1):

$$
\begin{aligned}
cost_{\text{label}} &= \begin{cases} 0 & \text{if detection labels are the same} \\ \text{Cost}_{\max} & \text{otherwise} \end{cases} \\
cost_{\text{IoU}} &= \begin{cases} \frac{1}{\text{IoU}} & \text{if IoU} > \frac{1}{\text{Cost}_{\max}} \\ \text{Cost}_{\max} & \text{otherwise} \end{cases} \\
cost_{\text{total}} &= cost_{\text{label}} + cost_{\text{IoU}}
\end{aligned} \tag{1}
$$

The total cost is the sum of the label-based cost and the cost derived from the overlap between bounding boxes, calculated using Intersection over Union (IoU).

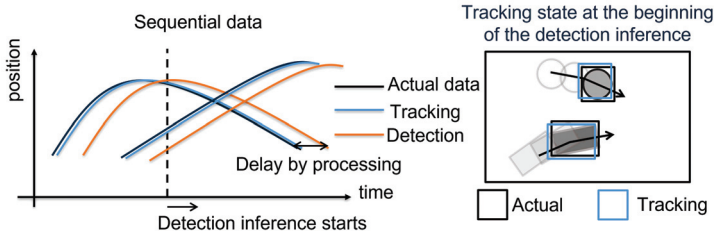


Figure 5. (**left**) shows sequential data; the black line represents the ground truth, the blue line indicates the tracked data from the tracking thread, and the orange line shows the detection data from the detection thread. The horizontal offsets between the blue and orange lines relative to the black line illustrate the delays caused by processing time. The vertical dashed line marks the moment when deep learning-based detection inference begins. (**right**) illustrates the tracking state corresponding to the time indicated by the dashed line in the left figure.

In conventional methods [12–14], tracker matching typically relies on predicted positions from Kalman Filters, which are effective only in scenarios with simple motion. In contrast, the proposed method improves matching accuracy by utilizing real-time tracking data from the tracking thread.

3.4. Hybrid Tracking

Figure 6 illustrates our hybrid tracking architecture inspired from [41]. New correlation filters (CFs) and templates from the detection thread are used to update the trackers only if the tracking score from the tracking thread is under threshold.

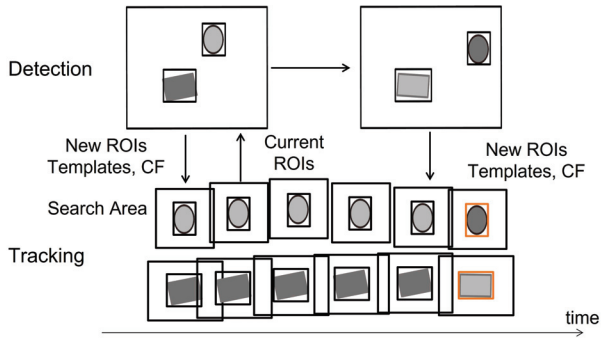


Figure 6. Overview of the hybrid tracking system combining deep learning-based low-frequency detection with classical high-frequency tracking methods. In the detection row, the large black box represents the entire image, while the smaller boxes around the objects indicate their ROIs. In the tracking row, the larger black boxes denote the search areas, and the smaller boxes represent the ROIs of the tracked objects. The horizontal axis indicates the time sequence. Orange boxes highlight the moments when the templates and correlation filters in the tracking thread are updated based on detection results.

To effectively define the search area around the target object when tracking is successful, the size of the search region, $(w_{\text{search}}, h_{\text{search}})$, is adaptively adjusted based on the velocity of the tracked object, as defined in Equation (2). In the case of the first detection or when the tracker has not been updated for more than 100 ms, both $scale_X$ and $scale_Y$ are set to the maximum value, $\text{scale}_{\text{max}}$.

$$\begin{aligned} w_{\text{search}} &= scale_{\text{X}} \cdot w_{\text{ROI}} \\ h_{\text{search}} &= scale_{\text{Y}} \cdot h_{\text{ROI}} \\ scale_{\text{X}} &= \max\left(\min\left(1 + \frac{\Delta x}{w_{\text{ROI}}}, \text{scale}_{\text{max}}\right), \text{scale}_{\text{min}}\right) \\ scale_{\text{Y}} &= \max\left(\min\left(1 + \frac{\Delta y}{h_{\text{ROI}}}, \text{scale}_{\text{max}}\right), \text{scale}_{\text{min}}\right) \end{aligned} \tag{2}$$

Here, $(w_{\text{ROI}}, h_{\text{ROI}})$ represent the current ROI's size, and Δx and Δy denote the horizontal and vertical movements from the previous detection, respectively. Both $scale_{\text{X}}$ and $scale_{\text{Y}}$ are constrained within the range [$\text{scale}_{\text{min}}$ (=1.5), $\text{scale}_{\text{max}}$ (=3.0)].

3.5. Handling Trackers

Methods for handling duplicate trackers and short-term occlusion are described in this section.

In crowded scenarios, multiple trackers may point to the same object, leading to redundant computations and increased processing time due to the larger number of trackers. To address this, we perform duplicate checking every time the tracking thread receives new data from the detection thread, when new trackers are likely to be generated. This procedure is based on two metrics: Intersection over Union (IoU) and velocity similarity. If both conditions in Equation (3) are satisfied, the redundant tracker is removed:

$$\begin{aligned} \text{IoU} &> \text{IoU}_{\text{duplication}} \\ \frac{\boldsymbol{v_i} \cdot \boldsymbol{v_j}}{|\boldsymbol{v_i}| \cdot |\boldsymbol{v_j}|} &> \cos(\theta_{\text{threshold}}) \end{aligned} \tag{3}$$

Here, $\text{IoU}_{\text{duplication}}$ is the threshold for determining duplication, and $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ denote the velocity of the respective trackers. The second condition in Equation (3) evaluates the directional similarity between the two trackers using the cosine of the angle between their velocity vectors.

When removing duplicate trackers, the system retains the tracker with the higher tracking score in terms of templates and correlation filters. As for the tracker itself, the one with the longer lifespan is preserved. This update strategy enables efficient tracking and contributes to extending the effective lifespan of valid trackers.

Secondly, to address short-term occlusions, we skip updating the correlation filter (CF) and template images in the tracking thread. During occlusion, we expect to see a discontinuity in the tracking score and movement. Therefore, if a tracker detects one of the discontinuities defined in Equation (4), the CF and template image are not updated. Instead, we only update the tracker's position based on Kalman Filter prediction.

$$\begin{aligned} \Delta\text{trackingScore} &\geq \Delta\text{trackingScore}_{\text{threshold}} \\ \frac{\boldsymbol{v_{t-1}} \cdot \boldsymbol{v_t}}{|\boldsymbol{v_{t-1}}| \cdot |\boldsymbol{v_t}|} &\leq \cos(\theta_{\text{threshold}}) \end{aligned} \tag{4}$$

Here, ΔtrackingScore represents the change in tracking score, $\Delta\text{trackingScore}_{\text{threshold}}$ represents the threshold for this change, $\boldsymbol{v_{t-1}}$ and $\boldsymbol{v_t}$ denote the velocity at times $(t-1)$ and t, respectively, and $\cos(\theta_{\text{threshold}})$ represents the threshold for changes in direction. The

first condition assesses an abrupt appearance change, and the second condition assesses an abrupt direction change. The consecutive skipping is up to $\text{Count}_{skipping}$.

4. Experiment

The demonstration video of our experiments can be found here [17].

4.1. Settings

We conducted experiments using a camera (XIMEA, Münster, Germany) set to 400 fps. Since the proposed method relies on classical high-speed detection and requires a high-frame-rate video, we collected our own dataset rather than using a public one. We used a PC as follows:

- CPU: AMD Ryzen 5 7645HX with Radeon Graphics;
- GPU: NVIDIA GeForce RTX 4050 Laptop GPU.

For all results presented in this paper, we compare each method and perform offline analyses. The experimental settings are illustrated in Figure 7.

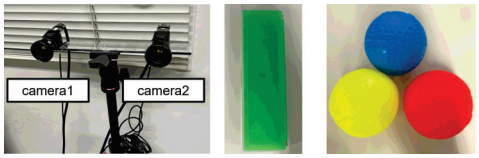


Figure 7. Experiment settings. (**left**) illustrates the cameras used to capture videos. (**right**) illustrates objects, plastic case and balls.

4.2. YOLO Training

For multiple object tracking, we used the YOLOv8n model [52] for detections and trained it with a customized dataset, which includes a plastic box and balls, as depicted in Figure 7. We used 6500 images for training. We used the YOLOv8-x detector for reference data.

4.3. Multiple Object Tracking

We conducted five tracking experiments for 2 objects (2 accelerated crossing objects, 2 categories, including a plastic case and ball, thrown balls), 3 objects, and 4 objects. The purposes are as follows:

- Two accelerated crossing balls: Evaluate tracking accuracy during abrupt motion changes and assess prevention of tracker switching.
- Two categories (plastic box and ball): Check whether the system is applicable to multi-category tracking and robust to rotational variations.
- Thrown balls: Assess the system's adaptability to changes in object scale and background conditions.
- Three and four balls: Evaluate the system's capability to track more than two objects, tracking accuracy in dense environments, and robustness to complete occlusion.

The hyperparameters used for this comparison are listed in Table 2.

We compared the proposed system with seven state-of-the-art tracking modules implemented based on BoxMOT [53]. From the slowest to the fastest in terms of processing speed, the tracking modules include Deep OC-SORT [48], StrongSORT [15], BoostTrack++ [49], Imprassoc [51], BoT-SORT [47], ByteTrack [13], and OC-SORT [14]. All tracking methods used a pretrained YOLOv8n model for detection. Except for BoT-SORT, ByteTrack, and OC-SORT, we employed a Re-ID model based on the Omni-Scale Network (OSNet) [54] to extract appearance features.

Table 2. Hyperparameters in the proposed MOT system.

Parameter		Value
Tracker lifespan	$\text{Counter}_{\text{valid}}$	4
	$\text{Counter}_{\text{lost}}$	5
Tracking score	PSR_{min}	5.0
	ZNCC_{min}	0.0
Duplicate Trackers (Equation (3))	$\text{IoU}_{\text{duplication}}$	0.6
Skip updating (Equation (4))	$\Delta\text{trackingScore}_{\text{threshold}}$	2.0
	$\cos(\theta_{\text{threshold}})$	$\pi/2$
	$\text{Count}_{\text{skipping}}$	3

We evaluated tracking accuracy using five metrics: Intersection over Union (IoU), mean Average Recall (mAR, Equation (5)) mean Average Precision (mAP, Equation (6)), Multiple Object Tracking Accuracy (MOTA, Equation (7)) [55], IDF1 (Equation (8)) [56], and Higher Order Tracking Accuracy (HOTA, Equation (9)) [57] including Detection Accuracy (DetA) and Association Accuracy (AssA).

$$\text{mAR} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|} \tag{5}$$

$$\text{mAP} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|} \tag{6}$$

$$\text{MOTA} = 1 - \frac{|\text{FN}| + |\text{FP}| + |\text{IDSW}|}{|\text{gtDet}|} \tag{7}$$

$$\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5 \cdot |\text{IDFN}| + 0.5 \cdot |\text{IDFP}|} \tag{8}$$

$$\begin{aligned} \text{HOTA} &= \frac{1}{19} \cdot \Sigma_{\alpha \in \{0.05,\ldots,0.95\}} \text{HOTA}_\alpha \\ \text{HOTA}_\alpha &= \sqrt{\text{DetA}_\alpha \cdot \text{AssA}_\alpha} \\ \text{DetA} &= \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|} \\ \text{AssA} &= \frac{\Sigma_{c \in \text{TP}} \text{Ass-IoU}(c)}{|\text{TP}|} \\ \text{Ass-IoU} &= \frac{|TPA|}{|TPA| + |FNA| + |FPA|} \end{aligned} \tag{9}$$

Here, |TP|, |IDTP|, and |TPA| represent the number of true positive trackers; |FN|, |IDFN|, and |FNA| indicate the number of false negative trackers; |FP|, |IDFP|, and |FPA| denote the number of false positive trackers; and |gtDet| is the number of ground truth trackers. |IDSW| indicates the number of tracker ID switches between consecutive frames, and α denotes the IoU threshold for detection, ranging from 0.05 to 0.95 in increments of 0.05. In our setting, MOTA, IDF1, and HOTA are averaged over all α values.

To calculate mAP, MOTA, and DetA, we performed frame-level matching between the predicted and ground truth trackers using the IoU and the Hungarian algorithm. For IDF1 and AssA, tracker pairs were determined by applying the Hungarian algorithm to minimize the total number of false positive and false negative trackers across all possible associations.

4.3.1. Tracking of 2 Accelerated Balls

According to Table 3, the proposed method operated at 470 fps, which was over 4.7 times faster than existing methods, and outperformed all conventional approaches

across all metrics for both detection and association accuracy. As shown in the estimated center positions and IoU transitions in Figure 8a,b, when the two balls began accelerating around 300 ms, the proposed method was able to track both objects with high precision and lower latency compared to ByteTrack and OC-SORT. As a result, OC-SORT lost track of the objects when the two balls crossed and one ball was partially occluded. These results demonstrate that the proposed method is robust to abrupt motion changes and partial occlusions and capable of accurately tracking fast-moving objects.

On the other hand, in the metric transitions shown in Figure 8d, the proposed method exhibited slightly lower scores than ByteTrack and OC-SORT at higher IoU thresholds. This drop was primarily due to false detections from the classical detector, which was observed in the first tracker from around t = 100∼300 ms in Figure 8, which deviated from the ground truth. This implied that the proposed method requires frequent updates by deep learning-based detections due to its limited robustness in long-term tracking.

Moreover, the bottom panels of Figure 8c,d show that processing speed has a more significant impact than deep learning-based appearance features on both detection and association performance.

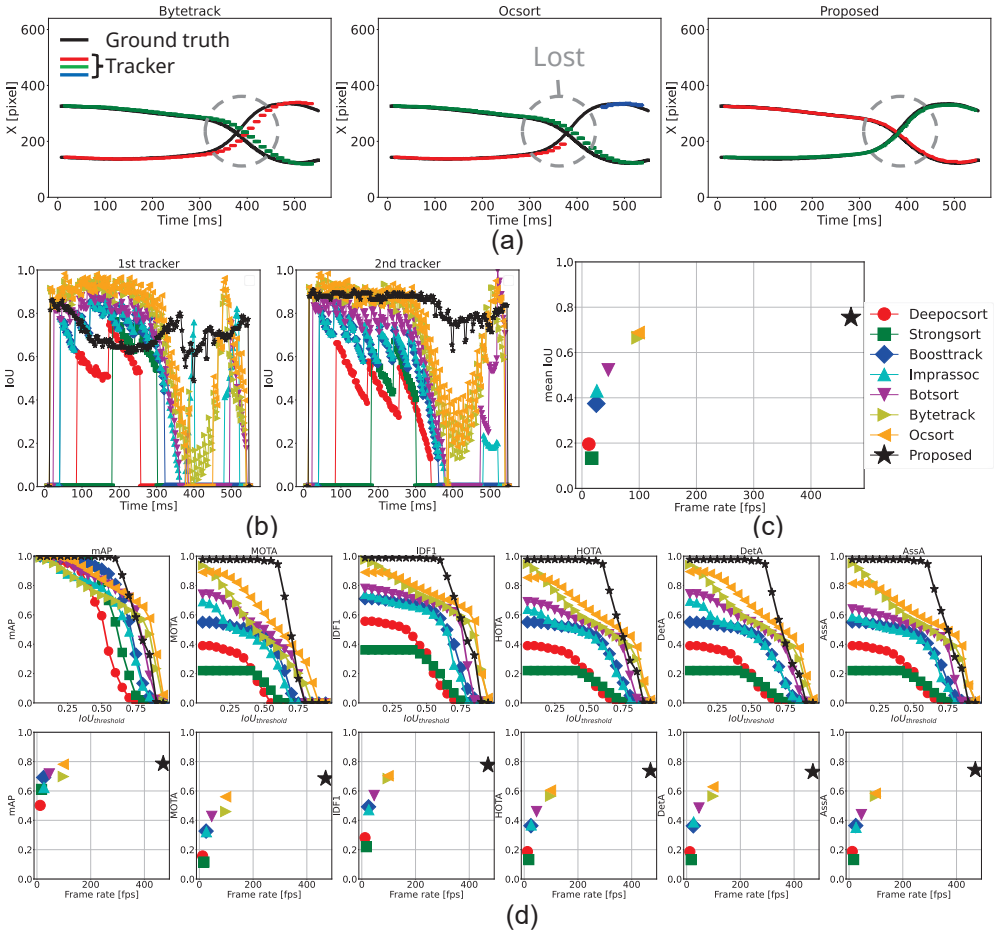


Figure 8. Tracking results for two accelerated balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. Plots with colors, but black represents each estimated trackers. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics including mAP, MOTA, IDF1, HOTA, DetA, and AssA. Bottom row: Relationship between frame rate and each performance metric.

Table 3. Tracking results for two accelerated balls with mean IoU, mAR, mAP, MOTA, IDF1, HOTA, DetA and AssA. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU			mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	Average							
deepocsort	12	0.184	0.206	0.195	0.196	0.502	0.157	0.282	0.187	0.186	0.188
strongsort	17	0.149	0.118	0.133	0.134	0.611	0.116	0.22	0.132	0.132	0.133
boosttrack	25	0.381	0.368	0.374	0.381	0.692	0.327	0.492	0.363	0.362	0.364
imprassoc	26	0.467	0.394	0.43	0.434	0.623	0.322	0.472	0.368	0.389	0.352
botsort	46	0.51	0.537	0.523	0.532	0.716	0.427	0.57	0.46	0.484	0.439
bytetrack	98	0.65	0.68	0.665	0.673	0.698	0.46	0.684	0.565	0.565	0.565
ocsort	98	0.665	0.71	0.688	0.699	0.782	0.56	0.704	0.605	0.628	0.584
proposed	**470**	**0.686**	**0.823**	**0.755**	**0.767**	**0.785**	**0.685**	**0.776**	**0.736**	**0.73**	**0.744**

4.3.2. Two Categories of Object Tracking

According to Table 4, the proposed method operated faster and outperformed all conventional approaches across all metrics in ball tracking. However, for box tracking, BoT-SORT, ByteTrack, and OC-SORT achieved better performance in both detection and association. This discrepancy was primarily due to the proposed method's lower detection accuracy and its inability to adapt to dynamically changing object shapes. One main factor was the reduced inference speed of deep learning-based detectors when multiple modules were executed simultaneously.

Table 4. Tracking results for ball and box tracking. The data are presented in the format of "ball's metric/box's metric". The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU	mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
deepocsort	12	0.01/0.021	0.007/0.02	0.06/0.377	0.002/0.018	0.012/0.035	0.007/0.019	0.007/0.02	0.007/0.018
strongsort	17	0.065/0.103	0.062/0.094	0.256/0.137	0.041/0.031	0.1/0.107	0.059/0.071	0.059/0.073	0.059/0.069
boosttrack	25	0.131/0.071	0.126/0.07	0.334/0.307	0.091/0.02	0.183/0.123	0.118/0.066	0.118/0.064	0.118/0.069
imprassoc	26	0.254/0.26	0.251/0.254	0.424/0.225	0.172/0.034	0.252/0.171	0.193/0.12	0.224/0.158	0.18/0.119
botsort	46	0.341/0.493	0.334/0.496	0.345/0.409	0.157/0.222	0.339/0.37	0.253/0.3	0.254/0.346	0.253/0.291
bytetrack	98	0.57/**0.594**	0.575/**0.6**	0.589/0.445	0.42/0.268	0.581/**0.504**	0.5/**0.404**	0.501/0.407	0.5/**0.409**
ocsort	98	0.609/0.563	0.615/0.571	0.63/**0.539**	0.451/**0.308**	0.622/0.489	0.538/0.391	0.538/**0.43**	0.537/0.384
proposed	**460**	**0.675**/0.42	**0.684**/0.423	**0.691**/0.373	**0.583**/0.137	**0.686**/0.314	**0.632**/0.248	**0.633**/0.286	**0.631**/0.222

As shown by the red plots in Figure 9a, ByteTrack and OC-SORT successfully tracked the box, while the red and blue plots from the proposed method failed to do so. Notably, around $t = 300$ ms, the YOLOv8 model failed to detect the box, resulting in gaps in the black plots. In contrast, the green plots from the proposed method interpolated the missing data and continued tracking. However, during this period, the tracker was unable to accurately adjust the bounding box shape, leading to a drop in IoU, as shown in Figure 9b.

Similarly, in the experiment with two accelerated balls, Figure 9c,d demonstrated a clear positive correlation between processing speed and tracking performance metrics.

In conclusion, this experiment confirmed that the proposed tracking system is capable of handling multi-category objects. However, for non-circular objects, high-frequency tracker updates using robust detection methods are essential. As future work, we plan to improve long-term robustness in tracking non-circular objects by both accelerating deep learning-based detection and developing a fast and rotation-invariant detection method.

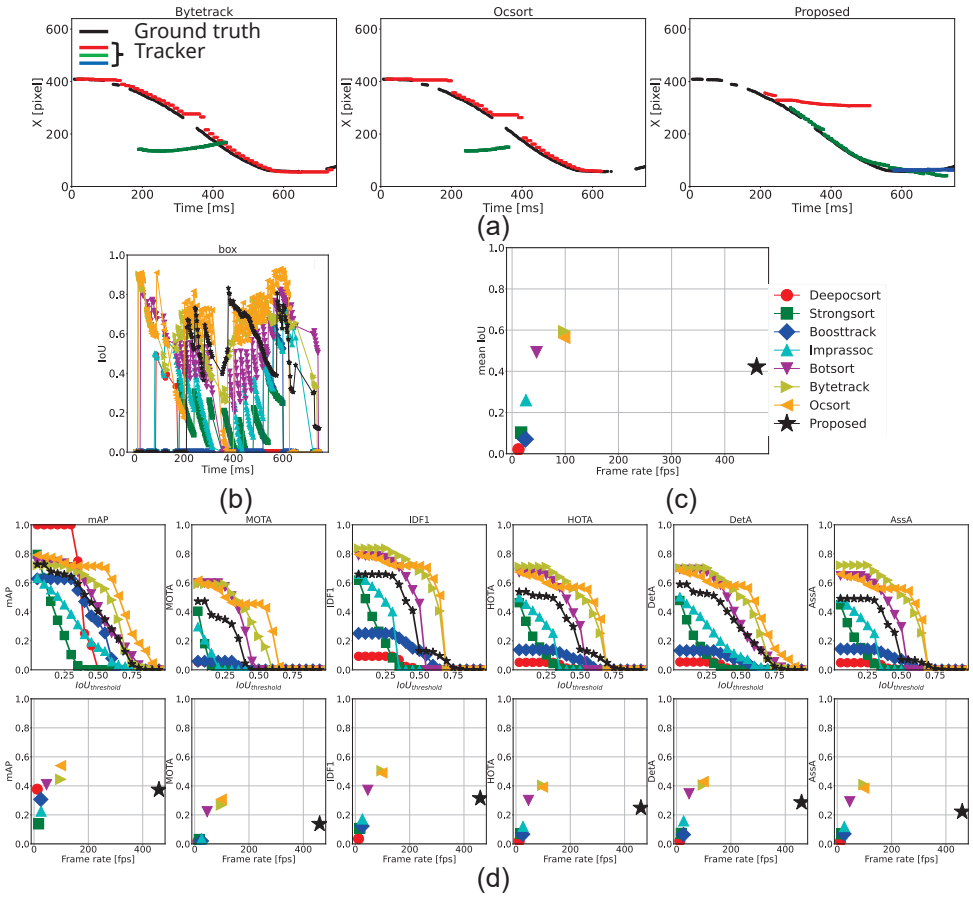


Figure 9. Tracking results for box tracking. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

4.3.3. Tracking of 2 Thrown Balls

According to Table 5, the proposed method operated faster; however, OC-SORT achieved higher values across all metrics. To understand the underlying cause, we refer to Figure 10. As shown by the red and green plots in Figure 10a, ByteTrack, OC-SORT, and the proposed method consistently tracked the same objects. However, the tracker in the proposed method slightly deviated from the ground truth center positions, which led to the initiation of additional trackers, visualized with blue and cyan plots. As seen in the IoU jump around $t = 200$ ms in Figure 10b, these duplicate trackers were eventually merged with the original ones through the duplication handling process described in Section 3.5. In the metric calculations, however, these duplicate trackers were counted as false positives in both detection and association, resulting in lower overall scores. To address this limitation, using coverage over the smaller bounding box instead of IoU may offer a more effective criterion. Nevertheless, the duplication-check procedure must be carefully designed to balance precision and recall, especially in crowded environments. Resolving this trade-off remains an important direction for future work.

Considering the low scores of slow-processing methods in Table 5 and the effect of processing speed on tracking metrics shown in Figure 10c,d, we conclude that high-speed tracking is essential for real-time recognition in dynamic environments.

This experiment demonstrated that the proposed method is adaptable to changes in object scale and background. However, it still faces challenges in managing duplicate

trackers considering the precision–recall trade-off as well as ensuring robust and high-frequency tracker updates.

Table 5. Tracking results for two thrown balls. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU			mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	Average							
deepocsort	12	0	0	0	0	0	0	0	0	0	0
strongsort	17	0	0	0	0	0	0	0	0	0	0
boosttrack	25	0.049	0.105	0.077	0.075	0.234	0.013	0.114	0.065	0.064	0.065
imprassoc	26	0.102	0.209	0.156	0.149	0.231	0.045	0.181	0.116	0.115	0.118
botsort	46	0.247	0.344	0.295	0.288	0.329	0.139	0.308	0.227	0.226	0.229
bytetrack	98	0.473	0.567	0.52	0.521	0.535	0.378	0.528	0.456	0.455	0.457
ocsort	98	0.505	**0.601**	**0.553**	**0.557**	**0.573**	**0.423**	**0.565**	**0.497**	**0.495**	**0.498**
proposed	**434**	**0.508**	0.553	0.53	0.531	0.497	0.326	0.492	0.409	0.427	0.397

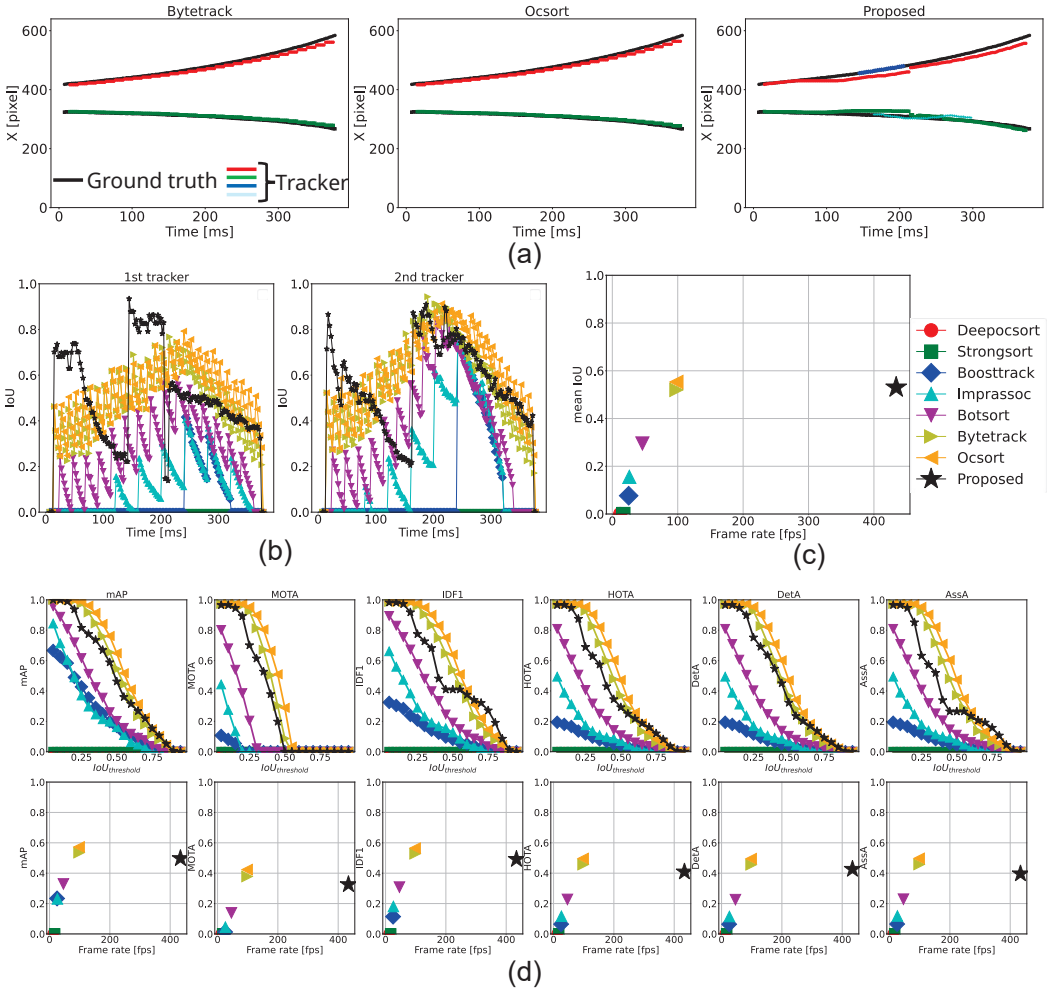


Figure 10. Tracking results for two thrown balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

4.3.4. Tracking of 3 Balls

According to Table 6, the proposed method operated at 243 fps, which is over 2.5 times faster than the conventional methods. ByteTrack and OC-SORT achieved higher scores in detection-related metrics such as mAP, MOTA, and DetA, while the proposed method outperformed them in association metrics, including IDF1 and AssA. As shown in Figure 11a,

the blue plots from ByteTrack and OC-SORT and the red plots from the proposed method represent the foremost ball. Among these, only the proposed method continuously tracked the same ball throughout the sequence, indicating its ability of continuous and respective tracking in crowded environments. However, similar to conventional methods, the proposed method still faced challenges in handling complete occlusions and duplicate trackers. To address complete occlusions, leveraging the classical high-speed detection method not only for fine-grained tracking but also for occlusion detection could be effective. Detecting occlusions through a drop in tracking scores can prevent trackers from switching to other objects, allowing them instead to rely on Kalman Filter predictions to wait for the occluded object to reappear. Alternatively, deploying a multi-camera system or extending to 3D tracking are promising options for improvement, which will be explored in future work.

As depicted in the figures, tracker switching in Figure 11a and lower association scores in Figure 11c,d, particularly in crowded environments, increased latency caused by slower processing speeds leads to greater deviations from the actual object positions. It results in an increase in the likelihood of tracker ID switches. Therefore, maintaining high processing speed is essential for accurate and consistent tracking.

This experiment demonstrated that the proposed method is capable of tracking more than two objects and that high-speed tracking significantly enhances discrimination and robustness in crowded environments.

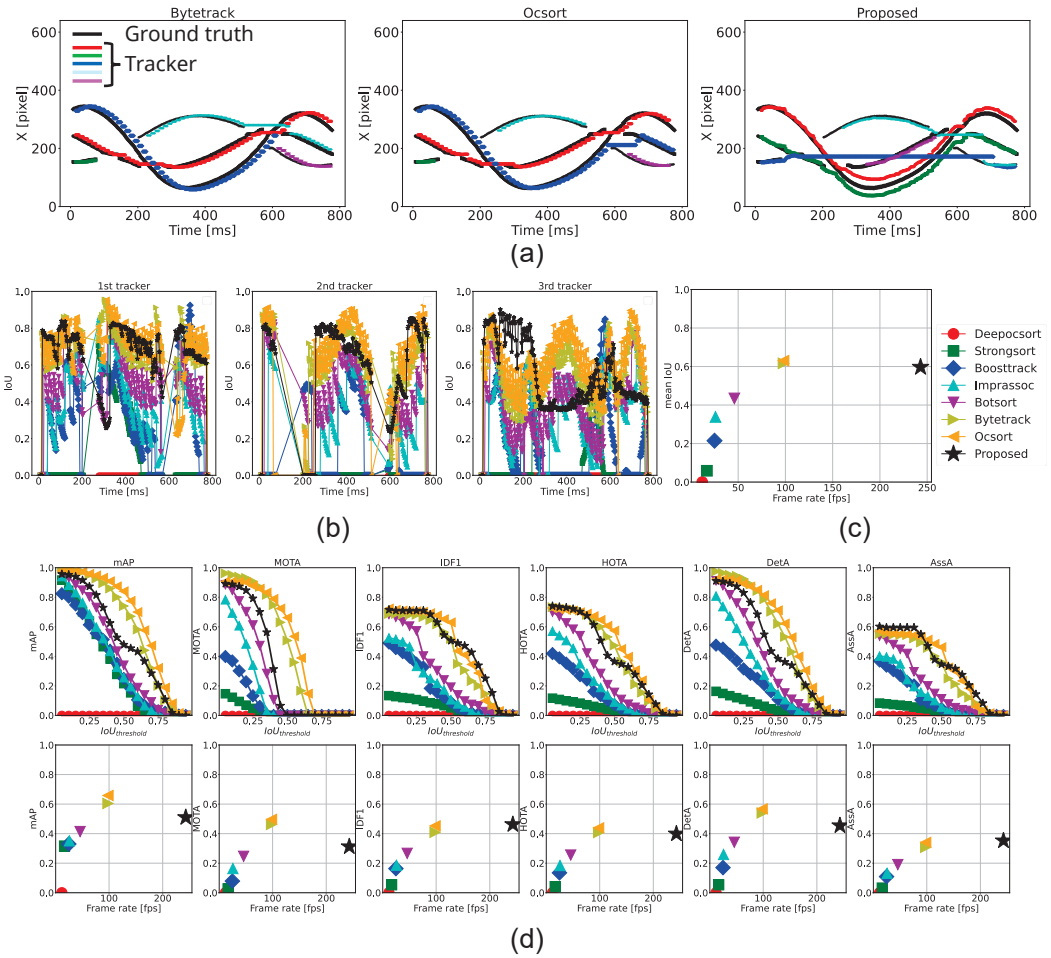


Figure 11. Tracking results for three balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

Table 6. Tracking results for three balls. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU				mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	3rd	Average							
deepocsort	12	0	0	0	0	0	0	0	0	0	0	0
strongsort	17	0.129	0.001	0.042	0.059	0.057	0.316	0.026	0.054	0.041	0.054	0.032
boosttrack	25	0.335	0.158	0.156	0.215	0.212	0.33	0.079	0.163	0.135	0.17	0.109
imprassoc	26	0.408	0.285	0.319	0.33	0.3338	0.351	0.164	0.186	0.184	0.259	0.132
botsort	46	0.503	0.434	0.377	0.434	0.432	0.414	0.247	0.267	0.255	0.342	0.191
bytetrack	98	**0.698**	**0.623**	0.544	0.617	0.624	0.604	0.462	0.409	0.407	0.54	0.308
ocsort	98	0.695	0.599	**0.6**	**0.63**	**0.639**	**0.658**	**0.497**	0.453	**0.439**	**0.567**	0.341
proposed	**243**	0.668	0.596	0.538	0.597	0.604	0.51	0.312	**0.462**	0.399	0.455	**0.352**

4.3.5. Tracking of 4 Balls

According to Table 7, the proposed method operated at 178 fps, which is over 1.8 times faster than the conventional methods. In this experiment, a similar trend to the three-ball tracking case was observed; ByteTrack and OC-SORT achieved higher scores in detection-related metrics such as mAP, MOTA, and DetA, while the proposed method outperformed them in association metrics, including IDF1 and AssA. Notably, the proposed method also achieved the highest score in HOTA.

Table 7. Tracking results for four balls. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU					mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	3rd	4th	Average							
deepocsort	12	0.063	0.032	0.095	0.033	0.053	0.052	0.32	0.024	0.062	0.041	0.049	0.034
strongsort	17	0.088	0.004	0.168	0.142	0.094	0.09	0.214	0.022	0.071	0.056	0.075	0.043
boosttrack	25	0.231	0.188	0.304	0.112	0.205	0.2	0.316	0.072	0.074	0.082	0.16	0.043
imprassoc	26	0.323	0.361	0.426	0.184	0.321	0.315	0.338	0.129	0.104	0.12	0.238	0.061
botsort	46	0.401	0.441	0.565	0.304	0.422	0.421	0.444	0.239	0.233	0.224	0.341	0.148
bytetrack	98	**0.586**	0.625	0.717	0.503	0.604	0.61	0.575	0.402	0.278	0.302	0.503	0.183
ocsort	98	0.557	**0.661**	**0.717**	0.538	**0.615**	**0.623**	**0.61**	**0.407**	0.19	0.246	**0.511**	0.12
proposed	**178**	0.469	0.654	0.604	**0.591**	0.579	0.586	0.548	0.355	**0.31**	**0.322**	0.466	**0.226**

As shown in Figure 12a, the green plots from ByteTrack and OC-SORT and the red plots from the proposed method represent the foremost ball. None of the methods maintained consistent tracking throughout, but the proposed method successfully handled the first crossing, where the conventional methods failed. However, from $t = 700$ ms, the proposed method lost the ball. This was caused because the tracker was prevented from updating by the deep learning-based module because its tracking score exceeded the threshold, as described in Section 3.4. Since the frame-level (local) tracking score does not always ensure consistent object identity, incorporating a global metric to evaluate tracking consistency is necessary. One potential solution is a temporal trajectory-aware metric, such as the state variance of a Kalman Filter.

As depicted in the tracker switching results in Figures 12c,d, detection-related scores such as mean IoU, mAP, MOTA, and DetA exhibited an exponential saturation behavior, while association-related scores like IDF1, AssA, and HOTA showed a monotonically increasing trend. In the proposed method, detection accuracy primarily depends on deep learning-based detection, while association accuracy is attributed to the hybrid tracking mechanism. Assuming both scores follow similar trends with respect to processing speed, it is inferred that while deep learning-based detection degraded at high speeds, the hybrid tracker compensated for poor detections using classical high-speed tracking methods.

In summary, we verified that the proposed method runs faster than conventional methods when tracking four objects simultaneously. We also demonstrated the effectiveness

of hybrid tracking in improving association performance and confirmed the necessity of enhancing detection accuracy to achieve balanced overall tracking performance.

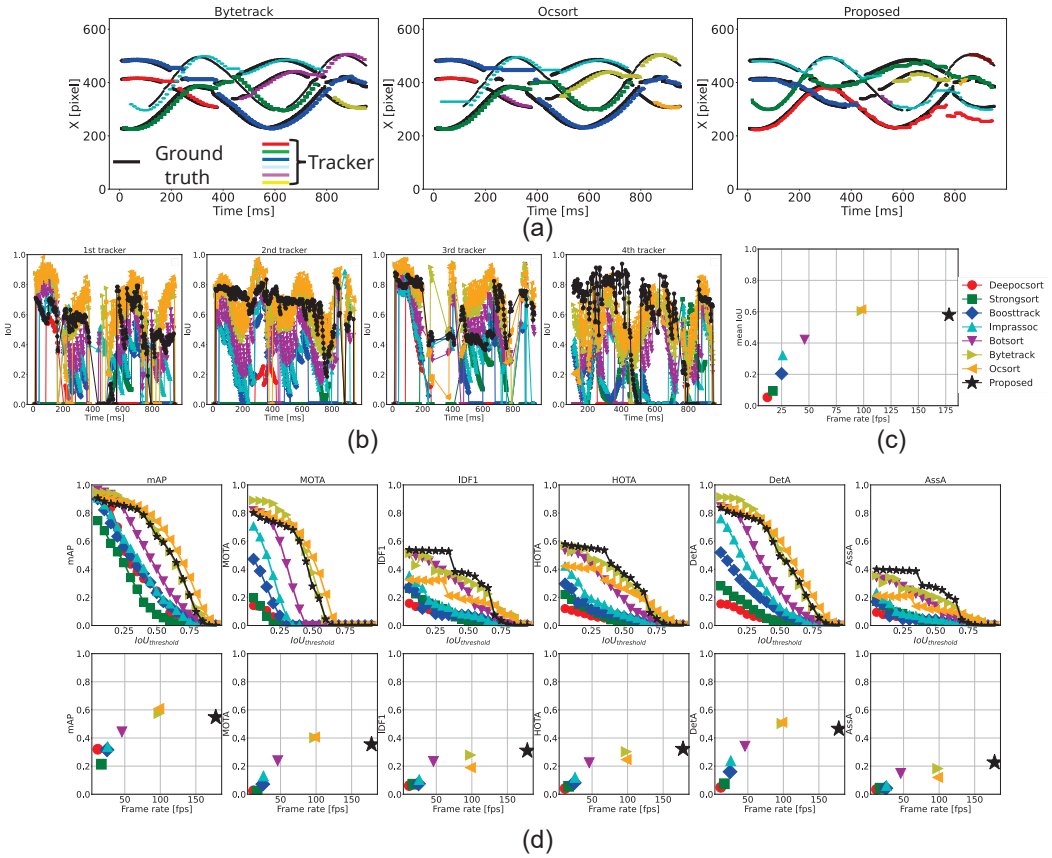


Figure 12. Tracking results for four balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

4.3.6. Summary of Tracking Experiments

We summarize the results of the five experiments in Table 8. Through these experiments, we confirmed that the proposed method achieved high-speed processing—up to 4.7× faster (470 fps) for two objects, 2.5× faster (243 fps) for three objects, and 1.8× faster (178 fps) for four objects, while maintaining versatility in terms of working environments, target types, and object quantities. The proposed method demonstrated superior performance across all metrics in the two-accelerated-ball experiment. However, it usually suffered from lower detection-related accuracy due to reduced deep learning-based detection performance caused by simultaneous multi-threaded operations. Despite this, our method showed better association performance than the conventional method, as indicated by metrics such as IDF1 and association accuracy (AssA). Therefore, by enhancing detection accuracy as shown in Table 8, the proposed system can achieve precise long-term tracking.

Furthermore, by comparing processing speed with each evaluation metric, ByteTrack and OC-SORT consistently achieved the best performance in MOTA, IDF1, and HOTA across all scenarios. We therefore conclude that in dynamically changing environments and when using a high-speed camera, processing speed has a more significant impact than deep learning-based appearance features on both detection and association performance.

Lastly, we briefly estimate the maximum objects where our method operates faster than the conventional methods. The processing speed of our system is primarily determined by the tracking time, $time_{tracking}$, of MOSSE and template matching, as preprocessing and postprocessing each take less than 50 µs. In our setup, $time_{tracking}$ per iteration was approximately 1.3 ms, based on averaged experimental processing time. Due to CPU core limitations, we applied MOSSE and template matching to each tracker sequentially rather than simultaneously. Therefore, assuming memory bandwidth is not a limiting factor, our system can operate faster than conventional methods when tracking seven or fewer objects.

Table 8. Summary of the five MOT experiments.

Experiment	Frame Rate (fps)	Achievements	Challenges and Future Work
2 Accelerated Balls (Section 4.3.1)	470	Robust to abrupt motion changes and partial occlusions, capable of accurately tracking fast-moving objects.	Requires high-frequency updates from deep learning-based detections for robust long-term tracking.
2 Categories (Section 4.3.2)	460	Capable of handling multi-category objects.	Improve tracking of non-circular objects by accelerating deep learning-based detection and developing a fast, rotation-invariant detection method.
2 Thrown Balls (Section 4.3.3)	434	Adaptable to changes in object scale and background.	Manage duplicate trackers while balancing the precision–recall trade-off, ensuring robust and high-frequency tracker updates.
3 Balls (Section 4.3.4)	243	Capable of tracking more than two objects with higher association accuracy.	Address complete occlusion by implementing fine-grained tracking-based occlusion detection and expanding to a multi-camera system.
4 Balls (Section 4.3.5)	178	Demonstrates the effectiveness of hybrid tracking in improving association performance.	Combine frame-level local tracking with trajectory-aware global scores to achieve long-term tracking.

5. Ablation Study

5.1. Effectiveness of Each Method

We confirmed the effects of each element of our proposed methods with a two-ball-crossing situation. The methods to be checked are as follows:

1. Tracking thread: Correlation filter-based tracking, MOSSE.
2. Tracking thread: Template matching.
3. Tracking thread: MOSSE + Template matching.
4. Database thread (Section 3.2): Compensation by Kalman prediction.
5. How to update trackers (Section 3.4): Check tracking score of the current templates.
6. Dynamic search area size adjustment based on velocity of trackers (Equation (2)).
7. Delete duplicate trackers (Section 3.5).
8. Skip updating templates (Section 3.5).

From now on, the indices greater than 3 represent methods that build upon the previously listed methods by incorporating the technique corresponding to the given index. For example, Method 5 represents one combining techniques 3, 4 and 5. Additionally, for methods prior to Method 6, the search area size is fixed at three times the size of the current ROI. We evaluated the contribution of each method in terms of processing speed and tracking accuracy.

The results are presented in Table 9. First, we compared Methods 1 to 3: "MOSSE", "Template Matching", and "MOSSE + Template Matching". Among these, Method 2 achieved the best performance in both tracking accuracy and processing speed, with a MOTA of 0.538, IDF1 of 0.539, and HOTA of 0.52. However, as shown in the top-left three panels of Figure 13a, multiple colors appear within the same object, indicating that none of these methods could handle long-term tracking. For subsequent methods, we adopt

Method 3 ("MOSSE + Template Matching"), as MOSSE is known for its robustness to scale and rotation variations.

Table 9. Tracking results for two balls. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU			mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	Average							
1	612	0.459	0.409	0.435	0.44	0.586	0.259	0.297	0.258	0.365	0.184
2	**710**	**0.812**	0.57	0.696	0.709	**0.733**	0.538	0.539	0.52	0.609	0.448
3	612	0.678	0.59	0.636	0.645	0.619	0.411	0.371	0.355	0.513	0.248
4	284	0.748	0.67	**0.711**	**0.723**	0.503	0.248	0.335	0.316	0.461	0.218
5	334	0.691	0.717	0.703	0.715	0.512	0.304	0.525	0.446	0.477	0.419
6	340	0.614	**0.745**	0.677	0.686	0.509	0.301	0.396	0.354	0.47	0.27
7	442	0.689	0.655	0.673	0.682	0.687	0.574	0.532	0.516	0.624	0.428
8	434	0.753	0.662	0.71	**0.723**	0.722	**0.611**	**0.719**	**0.656**	**0.653**	**0.661**

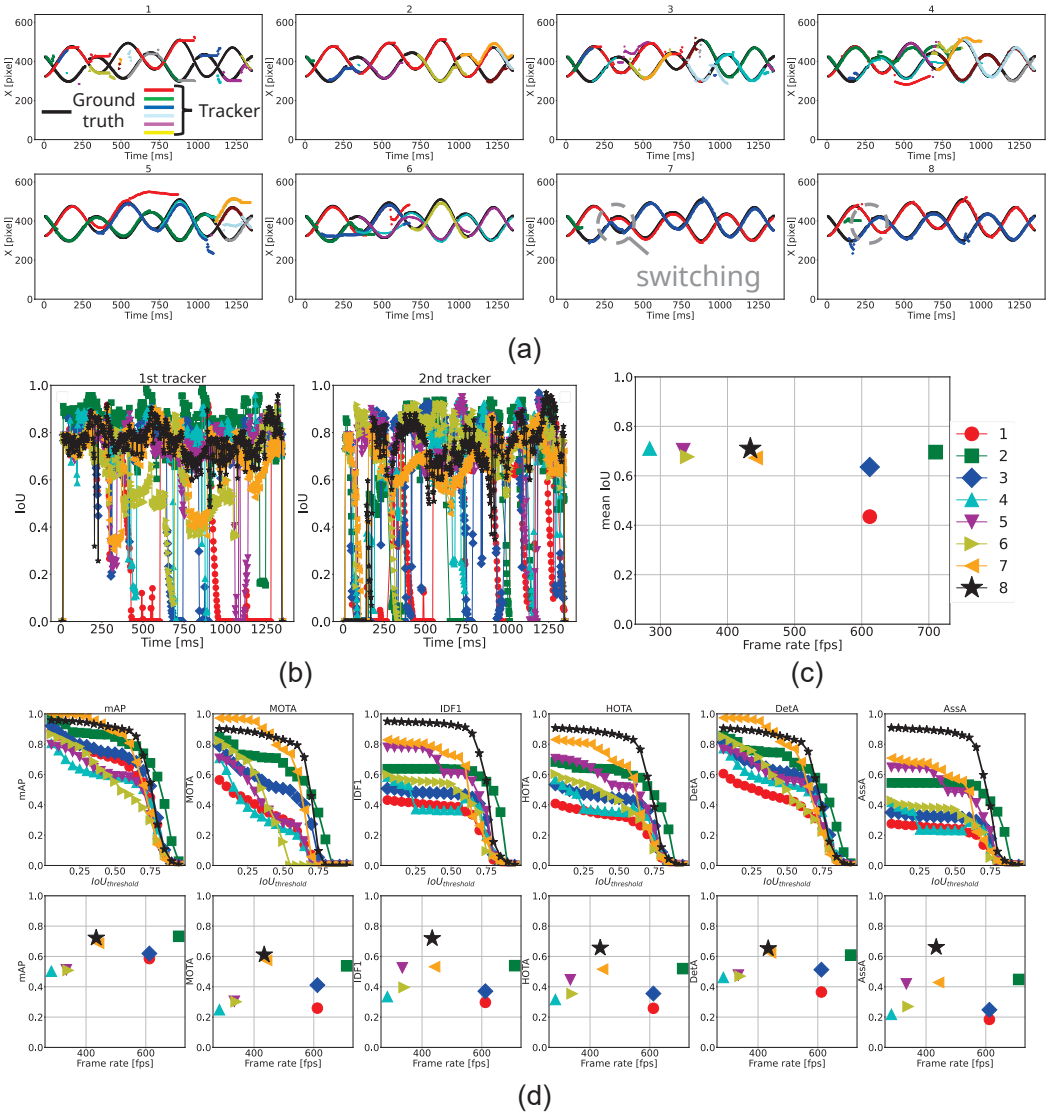


Figure 13. Tracking results for two balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by 8 methods. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

Comparing Method 4 with Method 3 in Table 9, the integration of Kalman prediction enables the system to maintain object tracking over a longer duration and achieves the best average IoU. In the top-right panel of Figure 13a, black-colored plots are not visible except during t = 50∼200 ms, where other colors appear, indicating an improved metric, recall. This can be confirmed its highest score in mAR at 0.723 from Table 9. However, the number of trackers assigned to the same ball increased, resulting in degraded MOTA, IDF1, HOTA, and processing speed.

In Method 5, a tracking-score-based update procedure was introduced to prioritize the use of recent, high-confidence templates over obsolete deep learning-based ones. As shown in the bottom-left panel of Figure 13a, the length of the same-colored plots increased compared to Method 4, suggesting more consistent tracking and a decrease in redundant trackers. Consequently, as confirmed in Table 9, processing speed slightly improved while maintaining appropriate templates. However, we still see some duplicate trackers remaining, and performance in metrics other than average IoU was still lower than the best-performing method (Method 2).

According to the results of Method 6 in Table 9, dynamically adjusting the search area size slightly improved the processing speed to 340 fps but did not enhance tracking performance. Nevertheless, for other scenarios, we adopted this function to enable efficient exploration by reducing unnecessary searches.

In Method 7, as shown in Table 9 and the seventh panel of Figure 13a, removing duplicate trackers significantly improved the processing speed to 442 fps and extended tracker lifespan by merging duplicate trackers. As a result, Table 9 showed that Method 7 achieved the highest MOTA score of 0.574 and competitive IDF1 and HOTA scores of 0.532 and 0.516, respectively. As shown in the top panels of Figure 13d, all metrics for Method 7 exceeded those of Method 2 at an $\mathrm{IoU}_{\mathrm{threshold}}$ of 0.5. However, at higher $\mathrm{IoU}_{\mathrm{threshold}}$ values, Method 2 achieved better scores, resulting in comparable overall performance in IDF1 and HOTA.

In the eighth row of Figure 13a, the introduction of update skipping successfully prevented a tracker ID switch around $t = 300$ ms, an issue unresolved in Method 7. Furthermore, consistent color plots in Figure 13a, stable IoU values exceeding 0.6 in Figure 13b, and the highest scores in each metric by $\mathrm{IoU}_{\mathrm{threshold}} = 0.75$ shown in Figure 13b indicate that both balls were reliably tracked in terms of detection and association. Although the tracker occasionally deviated from the ground truth around t = 900 ms, as shown in Figure 13a, due to update skipping, it successfully recovered to the correct position through deep learning-based updates. As shown in Table 9, Method 8 (our proposed method) achieved the best overall performance, with a MOTA of 0.611, IDF1 of 0.719, and HOTA of 0.656.

Finally, the contributions of each component are summarized in Table 10.

Table 10. Effects of each proposed MOT technique on processing speed and tracking accuracy, recall, precision and association. The upward arrow indicates an improvement in the score, while the downward arrow indicates a deterioration.

Method	Processing Speed	Recall	Precision	Association
MOSSE + Template matching	-	↑	↑	↑
Kalman Filter	↓	↑	-	-
Check tracking scores	-	-	↑	↑
Dynamic search area size adjustment	↑	-	-	-
Delete duplicate trackers	↑	-	↑	↑
Skip updating trackers	-	-	-	↑

5.2. Comparison with Other Methods

According to Table 11, the proposed method operated faster than existing methods, and outperformed all conventional approaches across all metrics for both detection and association accuracy. As shown in the estimated center positions in Figure 14a, ByteTrack and OC-SORT suffered from tracker ID switching around $t = 300$ ms and $t = 1000$ ms, while the proposed method overcame this. As shown in IoU and metric transitions in Figure 14b–d, the proposed method kept tracking with higher IoU, resulting in higher metrics over all $\mathrm{IoU}_{\mathrm{threshold}}$. These results demonstrate that the proposed method is effective in handling partial occlusions and long-term fast object tracking. As in other scenarios, the bottom panels of Figure 14c,d show that processing speed has greater influence than deep learning-based appearance features on both detection and association performance.

Table 11. Tracking results for two balls. The bold value indicates the best score for each metric.

Method	Frame Rate (fps)	IoU			mAR	mAP	MOTA	IDF1	HOTA	DetA	AssA
		1st	2nd	Average							
deepocsort	12	0	0	0	0	0	0	0	0	0	0
strongsort	17	0.007	0.046	0.026	0.026	0.369	0.011	0.041	0.023	0.025	0.022
boosttrack	25	0.022	0.004	0.013	0.012	0.11	0.001	0.01	0.007	0.011	0.005
imprassoc	26	0.252	0.236	0.244	0.237	0.224	0.033	0.039	0.055	0.15	0.021
botsort	46	0.305	0.294	0.3	0.295	0.354	0.129	0.115	0.122	0.231	0.065
bytetrack	98	0.496	0.494	0.495	0.495	0.489	0.331	0.279	0.275	0.418	0.182
ocsort	98	0.569	0.523	0.547	0.551	0.555	0.374	0.324	0.315	0.465	0.214
proposed	**434**	**0.753**	**0.662**	**0.71**	**0.723**	**0.722**	**0.611**	**0.719**	**0.656**	**0.653**	**0.661**

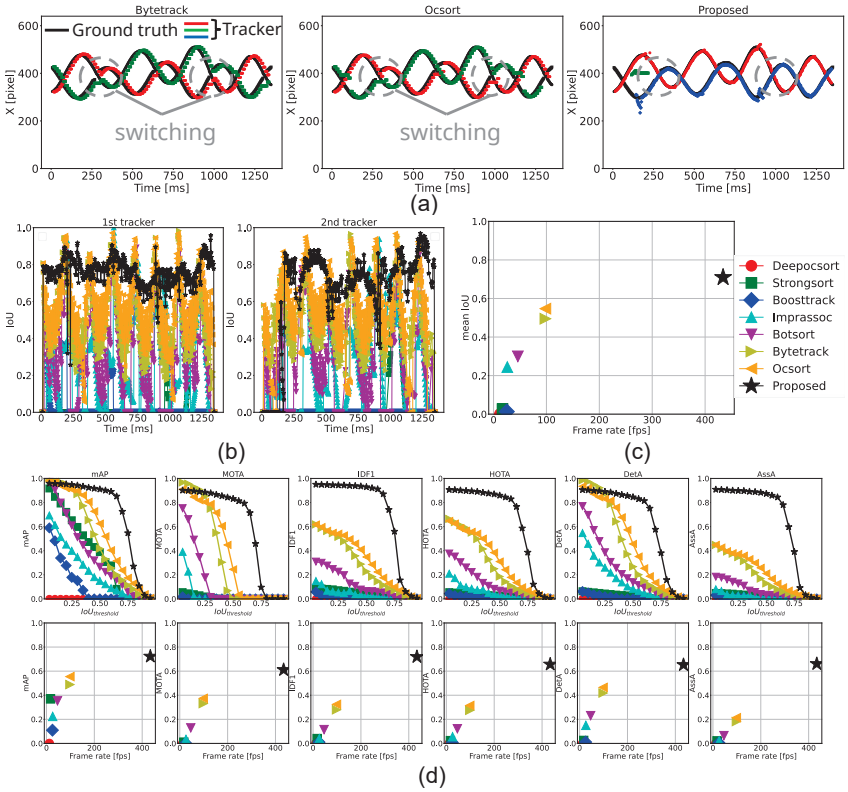


Figure 14. Tracking results for two balls. (**a**) Trajectories of the horizontal position (x-coordinate) of the ball centers tracked by ByteTrack, OC-SORT, and the proposed method. (**b**) Temporal transition of IoU for each ground-truth tracker. (**c**) Comparison of each method in terms of frame rate and mean IoU. (**d**) Top row: Relationship between IoU threshold and performance metrics. Bottom row: Relationship between frame rate and each performance metric.

6. Conclusions

Conventional tracking systems have struggled to satisfy all the essential criteria for a versatile tracking solution: low latency, high accuracy, and robustness across diverse environments and varying numbers and types of targets. To address this, we proposed a high-speed multiple object tracking (MOT) system that meets all of these requirements.

Our system is mainly composed of two components. The first is a hybrid tracking framework that integrates deep learning-based detection, correlation filter tracking, template matching-based detection, and Kalman prediction for motion compensation. The second is a detection label-based tracker management strategy that synchronously handles multiple trackers across asynchronous threads while maintaining low latency. To ensure robust and continuous tracking, we further proposed three techniques: dynamic search area adjustment, duplicate tracker elimination, and update skipping for occlusion-aware templates. These methods' effectiveness was verified through ablation studies.

We compared our system with seven conventional methods across six scenarios using recent MOT evaluation metrics, including IoU, mAR, mAP, MOTA, IDF1, and HOTA. Since our system leverages classical high-speed detection, it requires a high-speed camera—an aspect rarely addressed in prior work. Across all scenarios, we observed that in dynamically changing environments and when using a high-speed camera, processing speed has a more significant impact than deep learning-based appearance features on both detection and association performance.

Our experiments demonstrated the effectiveness of our system as follows. In terms of processing speed, we achieved up to 4.7× faster operation (470 fps) for two objects, 2.5× faster (243 fps) for three objects, and 1.8× faster (178 fps) for four objects. In the two-ball accelerated motion scenario and two-ball crossing motion, our system consistently achieved the best performance across all metrics. It also proved robust to partial occlusions and successfully mitigated tracker ID switches, which conventional methods struggled to overcome. In experiments involving two object categories (ball and box) and two thrown balls, our system demonstrated robustness to multiple categories and changes in rotation, scale, and background. However, conventional methods such as ByteTrack and OC-SORT outperformed our system in MOTA, IDF1, and HOTA under some conditions, largely due to the relatively lower detection accuracy in our framework. In three- and four-object tracking scenarios, our system showed strong performance, particularly in IDF1 and association accuracy (AssA), highlighting its ability to maintain robust associations despite lower detection accuracy.

While our system achieved comparable or superior tracking accuracy, it has three limitations and presents two key challenges for future work. First, as the system relies on classical high-speed detection, its effectiveness is demonstrated only with a high-frame-rate video. Second, to improve long-term accuracy for non-circular objects, we plan to enhance detection performance by accelerating deep learning-based methods and developing high-speed, rotation- and scale-invariant detection algorithms. Third, to increase robustness against occlusion, we will implement fine-grained occlusion detection using global and local context-aware tracking metrics, and extend the system to support multi-camera tracking.

Finally, we introduced a multi-processing architecture that demonstrated effectiveness for high-speed MOT. We hope that our proposed system contributes to further advancements in MOT and serves as an efficient baseline for systems that integrate multiple asynchronous modules.

Author Contributions: Conceptualization, Y.K. and Y.Y.; methodology, Y.K. and Y.Y.; software, Y.K.; validation, Y.K.; formal analysis, Y.K.; investigation, Y.K.; resources, Y.K.; data curation, Y.K.; writing—original draft preparation, Y.K.; writing—review and editing, Y.K. and Y.Y.; visualization, Y.K.; supervision, Y.Y.; project administration, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by JSPS KAKENHI Grant Number 24K22303.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, S.; Lee, S.; Park, H. Integration of tracking, re-identification, and gesture recognition for facilitating human–robot interaction. *Sensors* **2024**, *24*, 4850. [CrossRef] [PubMed]
2. Zhang, P.; Li, X.; He, L.; Lin, X. 3D multiple object tracking on autonomous driving: A literature review. *arXiv* **2023**. arXiv:2309.15411. [CrossRef]
3. Advitiya, C.S.; Shenoy, A.R.; Shravya, A.R.; Battula, A.; Raghavendra, A.; Krishnan, R. Multiple object tracking for video analysis and surveillance: A literature survey. *Int. J. Innov. Sci. Res. Technol.* **2023**, *8*, 1617–1626. [CrossRef]
4. Cui, Y.; Zeng, C.; Zhao, X.; Yang, Y.; Wu, G.; Wang, L. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. *arXiv* **2023**, arXiv:2304.05170. [CrossRef]
5. Dhont, J.V.; Vandemeulebroucke, J.; Cusumano, D.; Boldrini, L.; Cellini, F.; Valentini, V.; Verellen, D. Multi-object tracking in MRI-guided radiotherapy using the tracking-learning-detection framework. *Radiother. Oncol.* **2019**, *138*, 25–29. [CrossRef]
6. Ishikawa, M. High-speed vision and its applications toward high-speed intelligent systems. *J. Robot. Mechatron.* **2022**, *34*, 912–935. [CrossRef]
7. Wright, M.; Nam, K.; Tong, J.; Muthu, S.; Andersson, L.; Nguyen, C. Improved safety and 3D scanning with human-robot collaboration. In Proceedings of the 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 27–29 November 2024; pp. 722–729. Available online: https://api.semanticscholar.org/CorpusID:276347639 (accessed on 18 April 2025).
8. Zhou, J.; Hirano, M.; Yamakawa, Y. High-speed recognition of pedestrians out of blind spot with pre-detection of potentially dangerous regions. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 945–950. [CrossRef]
9. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
10. Zhang, W.; Li, J.; Xia, M.; Gao, X.; Tan, X.; Shi, Y.; Huang, Z.; Li, G. OffsetNet: Towards efficient multiple object tracking, detection, and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 949–960. [CrossRef] [PubMed]
11. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
12. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016. [CrossRef]
13. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2022**, arXiv:2110.06864. [CrossRef]
14. Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; Kitani, K. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. *arXiv* **2023**, arXiv:2203.14360. [CrossRef]
15. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. StrongSORT: Make DeepSORT Great Again. *arXiv* **2023**, arXiv:2202.13514. [CrossRef]
16. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv* **2017**, arXiv:1703.07402. [CrossRef]
17. Demonstration Video. Available online: http://www.hfr.iis.u-tokyo.ac.jp/research/High-speed_MOT/index-e.html (accessed on 18 April 2025).
18. Hashemi, N.S.; Aghdam, R.B.; Ghiasi, A.S.B.; Fatemi, P. Template Matching Advances and Applications in Image Analysis. *arXiv* **2016**, arXiv:1610.07231. [CrossRef]
19. Chen, Z.; Hong, Z.; Tao, D. An Experimental Survey on Correlation Filter-based Tracking. *arXiv* **2015**, arXiv:1509.05520. [CrossRef]

20. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
21. Zhang, K.; Zhang, L.; Liu, Q.; Zhang, D.; Yang, M.-H. Fast Visual Tracking via Dense Spatio-temporal Context Learning. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 127–141. [CrossRef]
22. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
23. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef]
24. Ma, C.; Yang, X.; Zhang, C.; Yang, M.-H. Long-term correlation tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
25. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 749–758. [CrossRef]
26. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1981; Volume 2, pp. 674–679.
27. Farnebäck, G. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*; Bigun, J., Gustavsson, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 363–370.
28. Kroeger, T.; Timofte, R.; Dai, D.; Van Gool, L. Fast Optical Flow using Dense Inverse Search. *arXiv* **2016**, arXiv:1603.03590. [CrossRef]
29. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 FPS with deep regression networks. In *Computer Vision—ECCV 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 749–765. [CrossRef]
30. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional Siamese networks for object tracking. *arXiv* **2016**, arXiv:1606.09549. [CrossRef]
31. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-end representation learning for correlation filter based tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008. [CrossRef]
32. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with Siamese region proposal network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980. [CrossRef]
33. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 12–20 June 2019; pp. 4277–4286. [CrossRef]
34. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. *arXiv* **2020**, arXiv:1911.06188. [CrossRef]
35. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302. [CrossRef]
36. Nishimura, H.; Tasaka, K.; Kawanishi, Y.; Murase, H. Object tracking by multiple correlation filters robust to appearance changes. *J. Inst. Image Inf. Telev. Eng.* **2019**, *73*, 1004–1012. [CrossRef]
37. Sun, Z.; Chen, J.; Chao, L.; Ruan, W.; Mukherjee, M. A survey of multiple pedestrian tracking based on tracking-by-detection framework. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1819–1833. [CrossRef]
38. Liu, F.; Mao, K.; Qi, H.; Liu, S. Real-time long-term correlation tracking by single-shot multibox detection. *Opt. Eng.* **2019**, *58*, 013105. [CrossRef]
39. Wang, X.; Zhang, K.; Li, S.; Hu, Y.; Yan, J. An optimal long-term aerial infrared object tracking algorithm with re-detection. *IEEE Access* **2019**, *7*, 114320–114333. [CrossRef]
40. Shin, J.; Kim, H.; Kim, D.; Paik, J. Fast and robust object tracking using tracking failure detection in kernelized correlation filter. *Appl. Sci.* **2020**, *10*, 713. [CrossRef]
41. Matsuo, A.; Yamakawa, Y. High-Speed Tracking with Mutual Assistance of Feature Filters and Detectors. *Sensors* **2023**, *23*, 7082. [CrossRef]
42. Ultralytics. YOLOv5 in PyTorch [GitHub Repository]. 2022. Available online: https://github.com/Ultralytics/Yolov5 (accessed on 18 April 2025).
43. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. *arXiv* **2020**, arXiv:1909.12605. [CrossRef]

44. Meinhardt, T.; Kirillov, A.; Leal-Taixé, L.; Feichtenhofer, C. TrackFormer: Multi-Object Tracking with Transformers. *arXiv* **2022**, arXiv:2101.02702. [CrossRef]
45. Lv, W.; Zhang, N.; Zhang, J.; Zeng, D. One-shot multiple object tracking with robust ID preservation. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 4473–4488. Available online: https://api.semanticscholar.org/CorpusID:265859079 (accessed on 18 April 2025). [CrossRef]
46. Hou, H.; Shen, C.; Zhang, X.; Gao, W. CSMOT: Make one-shot multi-object tracking in crowded scenes great again. *Sensors* **2023**, *23*, 3782. [CrossRef]
47. Aharon, N.; Orfaig, R.; Bobrovsky, B.-Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv*. **2022**, arXiv:2206.14651. [CrossRef]
48. Maggiolino, G.; Ahmad, A.; Cao, J.; Kitani, K. Deep OC-SORT: Multi-Pedestrian Tracking by Adaptive Re-Identification. *arXiv* **2023**, arXiv:2302.11813. [CrossRef]
49. Stanojević, V.; Todorović, B. BoostTrack++: Using tracklet information to detect more objects in multiple object tracking. *arXiv* **2024**, arXiv:2408.13003. [CrossRef]
50. Stanojevic, V.D.; Todorovic, B.T. BoostTrack: Boosting the similarity measure and detection confidence for improved multiple object tracking. *Mach. Vis. Appl.* **2024**, *35*, 53. [CrossRef]
51. Stadler, D.; Beyerer, J. An Improved Association Pipeline for Multi-Person Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 3170–3179. [CrossRef]
52. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLO (Version 8.0.0) [Computer Software]. Ultralytics. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 18 April 2025.)
53. Broström, M. *BoxMOT: Pluggable SOTA Tracking Modules for Object Detection, Segmentation and Pose Estimation Models*, Version 12.0.7 [Software]; Zenodo: Geneva, Switzerland, 2024. [CrossRef]
54. Zhou, K.; Yang, Y.; Cavallaro, A.; Xiang, T. Omni-Scale Feature Learning for Person Re-Identification. *arXiv* **2019**, arXiv:1905.00953. [CrossRef]
55. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831. [CrossRef]
56. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In *Computer Vision—ECCV 2016 Workshops*; Springer: Cham, Switzerland, 2016; pp. 17–35.
57. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.H.S.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 548–578. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Improvement of SAM2 Algorithm Based on Kalman Filtering for Long-Term Video Object Segmentation

Jun Yin [1], Fei Wu [1], Hao Su [2,*], Peng Huang [2] and Yuetong Qixuan [2]

[1] School of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China; 12221242@zju.edu.cn (J.Y.); wufei@zju.edu.cn (F.W.)
[2] Zhejiang Dahua Technology Co., Ltd., Hangzhou 310053, China; huang_peng@dahuatech.com (P.H.); qi_xuanyuetong@dahuatech.com (Y.Q.)
* Correspondence: chaozhong2010@163.com

Abstract

The Segment Anything Model 2 (SAM2) has achieved state-of-the-art performance in pixel-level object segmentation for both static and dynamic visual content. Its streaming memory architecture maintains spatial context across video sequences, yet struggles with long-term tracking due to its static inference framework. SAM 2's fixed temporal window approach indiscriminately retains historical frames, failing to account for frame quality or dynamic motion patterns. This leads to error propagation and tracking instability in challenging scenarios involving fast-moving objects, partial occlusions, or crowded environments. To overcome these limitations, this paper proposes SAM2Plus, a zero-shot enhancement framework that integrates Kalman filter prediction, dynamic quality thresholds, and adaptive memory management. The Kalman filter models object motion using physical constraints to predict trajectories and dynamically refine segmentation states, mitigating positional drift during occlusions or velocity changes. Dynamic thresholds, combined with multi-criteria evaluation metrics (e.g., motion coherence, appearance consistency), prioritize high-quality frames while adaptively balancing confidence scores and temporal smoothness. This reduces ambiguities among similar objects in complex scenes. SAM2Plus further employs an optimized memory system that prunes outdated or low-confidence entries and retains temporally coherent context, ensuring constant computational resources even for infinitely long videos. Extensive experiments on two video object segmentation (VOS) benchmarks demonstrate SAM2Plus's superiority over SAM 2. It achieves an average improvement of 1.0 in J&F metrics across all 24 direct comparisons, with gains exceeding 2.3 points on SA-V and LVOS datasets for long-term tracking. The method delivers real-time performance and strong generalization without fine-tuning or additional parameters, effectively addressing occlusion recovery and viewpoint changes. By unifying motion-aware physics-based prediction with spatial segmentation, SAM2Plus bridges the gap between static and dynamic reasoning, offering a scalable solution for real-world applications such as autonomous driving and surveillance systems.

Keywords: SAM 2; long-term video; SA-V; LVOS; Kalman filter

1. Introduction

Recent advancements in computer vision have been revolutionized by segmentation models capable of precise pixel-level mask generation. The Segment Anything Model (SAM) [1] has emerged as a foundational paradigm for visual segmentation tasks due to

its exceptional ability to produce accurate instance masks. Building upon this framework, SAM2 [2] introduces a streaming memory architecture to sequentially process video frames, thereby maintaining spatial context over long sequences and significantly improving performance in video object segmentation (VOS) [2,3].

Despite these advancements, SAM2 exhibits systematic limitations in VOS scenarios, which severely restrict its utility in real-world dynamic environments. These limitations primarily arise from the model's inadequate capacity to model temporal dynamics, leading to inconsistencies in tracking over prolonged sequences. For instance, SAM2's reliance on spatial features causes positional drift and mask fragmentation when dealing with rapidly moving or self-occluded objects, as it fails to account for motion cues like velocity or trajectory changes. Similarly, its inability to explicitly model inter-object relationships—such as occlusion, proximity, or separation—results in confusion among visually similar targets in crowded scenes, while the linear decline in multi-object segmentation efficiency further hinders scalability. The streaming memory architecture, while preserving some historical context, lacks adaptive update mechanisms to handle long-term temporal dependencies, leading to error accumulation and eventual target loss in scenarios with drastic viewpoint changes or prolonged sequences [4,5]. Additionally, the model's poor sensitivity to fine-grained dynamic features, such as edge deformation or texture evolution during occlusion recovery, exacerbates segmentation inaccuracies. Although external cues like human annotations or supplementary trackers can partially alleviate these issues, they demand continuous manual intervention and fail to ensure temporal smoothness, particularly in sequences with abrupt motion or inconsistent mask quality [3]. The root cause of these challenges lies in the fundamental mismatch between SAM2's static segmentation framework and the dynamic requirements of long-term tracking, multi-target interactions, and evolving feature details in realistic video settings.

Kalman filtering (KF) [6] has long served as a foundational technique in visual object tracking (VOT), and is widely adopted in multi-object tracking (MOT) systems [7–10] due to its efficiency and interpretability. It excels in single-object tracking by modeling motion dynamics and reducing uncertainty through recursive state estimation. In MOT frameworks, KF is often combined with association methods such as JPDA [11] or MHT [12], and more recently with Labeled Random Finite Sets (LRFS) [13], to enhance robustness against occlusion and improve re-identification performance. However, these advanced MOT techniques typically involve complex association logic or require extensive offline optimization, which is not conducive to real-time applications or lightweight deployment. In contrast, we choose the linear Kalman filter as our baseline tracker, not only for its strong performance in motion prediction but also for its compatibility with streaming segmentation frameworks like SAM2 [2].

To address these limitations, we propose SAM2Plus, a training-free enhancement framework designed to improve SAM2's segmentation performance in VOS scenarios. SAM2Plus systematically integrates motion-aware prediction, adaptive evaluation, and scalable memory management to reconcile the static–dynamic inference inconsistency. The framework incorporates a Kalman filter prediction and adaptive update mechanism to anticipate object trajectories and dynamically adjust model states using physical motion constraints, thereby enhancing resilience to partial occlusions or sudden velocity changes. A dynamic thresholds and multi-criteria evaluation system balances model confidence scores with temporal tracking metrics (e.g., motion coherence, appearance consistency), enabling intelligent frame selection that prioritizes robust tracking information while adapting criteria to evolving environmental conditions. Furthermore, an adaptive streaming memory management module employs a novel dynamic library update strategy to selectively retain

high-quality historical context while discarding redundant entries, ensuring long-term tracking stability and preventing resource exhaustion, even in sequences with extreme temporal variability.

Experimental validation on two benchmark datasets—SA-V and LVOS—demonstrates the efficacy of SAM2Plus. Quantitatively, the proposed framework achieves an average improvement of 1.0 in J & F metrics across all 24 direct comparisons, with the highest enhancement of 2.3 in J & F scores observed in long-term sequences, as shown in Figure 1. These gains are realized without additional training or external parameters, as the Kalman filter introduces physical dynamics constraints to mitigate positional errors, dynamic thresholds reduce ambiguities in crowded scenes, and memory management ensures computational scalability. By unifying motion-aware prediction with SAM2's spatial segmentation strengths, SAM2Plus addresses the core limitations of its predecessor, providing a versatile solution for real-world applications such as surveillance, autonomous driving, and sports analytics.

The contributions of this work are threefold: first, a lightweight framework that enhances SAM2's segmentation capabilities without compromising its original performance; second, a systematic integration of motion prediction, adaptive evaluation, and memory management to bridge the gap between static segmentation and dynamic tracking; and third, empirical evidence of superior robustness in challenging scenarios, establishing a novel paradigm for long-term video object segmentation.

The remainder of this paper is organized as follows: Section 2 reviews the related work on VOS and SAM. Section 3 describes the proposed SAM2Plus method, including the integration of Kalman filtering, dynamic thresholds, and multi-metric evaluation, as well as memory management and optimization for long video sequences. Section 4 details the datasets, evaluation metrics, and results, demonstrating the effectiveness of SAM2Plus. Finally, Section 5 discusses the impact of the sub-modules, and Section 6 concludes the paper and outlines potential future work.

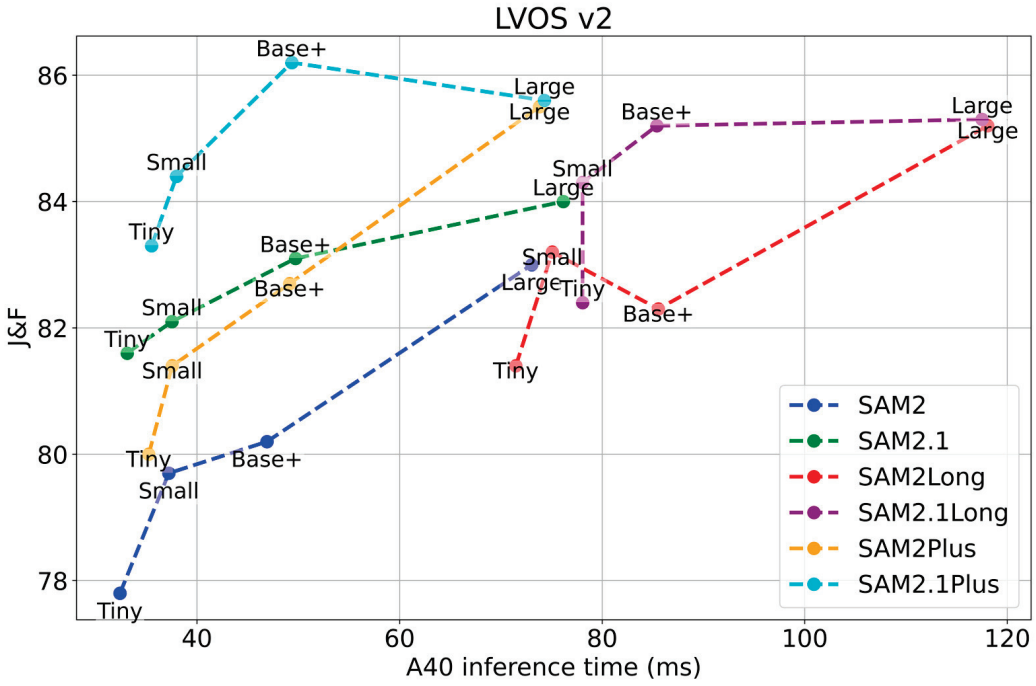


Figure 1. Comparative evaluation of SAM2Plus on the LVOS v2 dataset relative to advanced baselines.

2. Related Work

2.1. Visual Object Segmentation (VOS)

Video object segmentation (VOS) is a core computer vision task that performs pixel-level segmentation and tracking of target objects across video frames, requiring both spatial accuracy in individual frames and temporal consistency throughout sequences [3,14]. Existing approaches can be categorized by their supervision paradigms: Semi-supervised VOS methods, which dominate current research, utilize first-frame annotations as reference. The pioneering STM [15] introduced memory mechanisms for cross-frame matching, while CFBI [16] enhanced robustness through foreground–background feature contrast. Unsupervised approaches like MotionTrack [17] eliminate annotation requirements by leveraging motion cues, although with reduced precision. Traditional graph-based methods showed limitations in complex scenarios, leading to modern deep learning solutions spanning propagation-based OSVOS [18], detection-integrated TrackR-CNN [19], and transformer architectures SAM2 [2]. Current challenges include handling occlusions, appearance variations, and computational efficiency, with recent advances focusing on lightweight designs [3,20] and multimodal fusion [21]. Standard benchmarks (DAVIS [22], YouTube-VOS [23] and MOSE [24], SA-V [2]) and metrics (Jaccard, F-measure) drive progress in this evolving field [2,3]. This paper investigates the application of Segment Anything Model 2 (SAM 2) for semi-supervised VOS.

2.2. Segment Anything Model (SAM)

The Segment Anything Model (SAM) [1] is the first foundation model for general image segmentation tasks, aiming to achieve zero-shot open-world segmentation through prompt interaction. Some researchers have made lightweight designs of the original SAM in order to run SAM on resource-constrained embedded devices, thereby reducing the computing cost. FastSAM [25] accelerates the segmentation of all objects in the image by replacing the original transformer architecture with the CNN-based architecture YOLOv8-seg [26,27]. MobileSAM [28] introduces a decoupled distillation method to create a lightweight image encoder derived from SAM. MobileSAMv2 [29] significantly improves the efficiency of SAM by replacing grid search with object-aware prompt sampling. EfficientSAM [30] leverages the SAMI pretraining method to significantly reduce computational costs while maintaining performance comparable to large SAM models, making it more suitable for real-world applications. SAM-Lightening [31] accelerates SAM's vision backbone through dilated flash attention reconstruction coupled with adaptive inter-layer distillation, specifically targeting latency bottlenecks and memory-intensive computations in dense prediction systems. EfficientViT-SAM employs an EfficientViT-based image encoder to replace SAM's original heavy encoder, significantly improving segmentation efficiency while maintaining accuracy [32]. EdgeSAM [33] accelerates SAM for edge deployment by distilling its ViT-based image encoder into a CNN architecture, jointly optimizing the prompt encoder and mask decoder to preserve task-aware dynamics.

SAM2 [2] enhances Meta's original SAM [1] into a unified promptable architecture for both image and video segmentation, introducing a memory bank mechanism to retain object-specific features across frames for consistent single-object tracking, while preserving real-time processing and zero-shot generalization capabilities. Building on this foundation, Ma et al. adapted SAM2 to the medical domain by proposing MedSAM [34], a universal segmentation foundation model trained on 1.57 M multi-modal medical images. Chen et al. proposed SAM2-Adapter [35], the first adapter for Segment Anything 2 (SAM2) that achieves state-of-the-art performance in medical image segmentation, camouflaged object detection, and shadow detection tasks. While SAM2Long [4] enhances long video segmentation via tree-based memory, its computational overhead grows exponentially with higher FPS inputs and deeper memory architectures. SAMURAI [5] enhances tracking performance by

introducing a memory cleaning module that dynamically discards irrelevant information. Videnović et al. proposed SAM2.1++ [36], a distractor-aware introspective memory model that achieves significant performance gains in video object segmentation (VOS), outperforming existing methods on six of seven key benchmarks while introducing the DiDi dataset to quantitatively assess tracking stability. Jiang et al. proposed SAM2MOT [37], a Tracking-by-Segmentation paradigm for multi-object tracking that directly generates tracks from SAM2 masks, achieving state-of-the-art performance on DanceTrack (+2.1 HOTA) through novel trajectory management and cross-object interaction modules, while maintaining zero-shot generalization capability. We propose an enhanced framework based on SAM2 to further improve object segmentation capabilities in long videos.

3. Methods

As illustrated in Figure 2, we introduce SAM2Plus, an enhanced video segmentation framework built upon Segment Anything Model 2 (SAM2) [2]. The proposed architecture integrates three key improvements: the Kalman Filtering-Intersection Over Union (KF-IoU) fusion framework for temporal stability, the Adaptive Historical Frame Selection Strategy Based on Dynamic Threshold (AHFSS-DT) for efficient memory bank updates, and memory optimization techniques for long-term segmentation scalability.

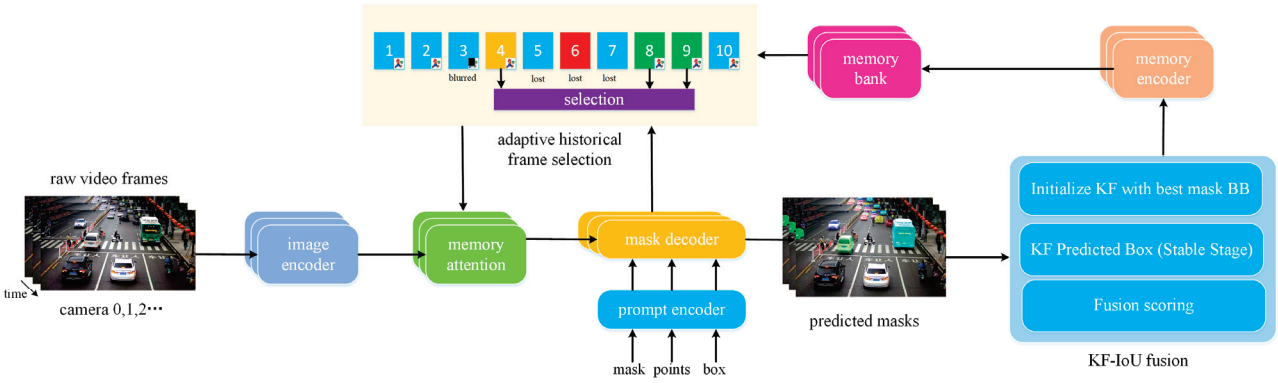


Figure 2. Overview of our SAM 2Plus architecture.

3.1. Baseline SAM2 Overview

We briefly review the SAM2 architecture to establish the foundation for our improvements. SAM2 [2] is a unified framework for interactive image and video segmentation, featuring four core components:

- Image Encoder: SAM2 employs a ViT-Hiera backbone [38] to extract hierarchical visual embeddings. These dense features are used for both image and video inference.
- Prompt Encoder: User-defined prompts (e.g., clicks, boxes, and masks) are encoded into task-specific embeddings to guide segmentation.
- Memory Bank: SAM2 maintains a memory bank of previous frame features to support video tracking. Features are encoded using a memory encoder and aligned through memory attention.
- Mask Decoder: Based on fused visual and memory features, the mask decoder generates segmentation masks conditioned on prompts. In video sequences, it also ensures temporal stability.

However, the fixed-window memory mechanism and the reliance on IoU-based mask selection in SAM2 result in two limitations: (1) reduced tracking accuracy in occlusion or fast-motion scenarios; and (2) sensitivity to single-frame segmentation errors that can propagate over time.

3.2. Kalman Filtering-IoU Fusion Framework

Given the success of SAMURAI [5] in the VOT task, to improve tracking consistency in VOS scenarios, we propose the Kalman Filtering-IoU (KF-IoU) fusion framework, which introduces a predictive component via Kalman Filtering to stabilize tracking and reduce the impact of single-frame errors while ensuring spatial segmentation precision through IoU optimization. The KF-IoU fusion framework combines the KF for motion estimation with the IoU prediction head of the mask decoder to dynamically refine the object's tracking state. Specifically, the KF module predicts the object's bounding box in the current frame based on prior motion dynamics, while the IoU head evaluates the confidence of the predicted mask. These two outputs are then fused through a weighted likelihood function, resulting in a more reliable mask proposal.

The Kalman filter [6] is an optimal recursive estimator for linear dynamical systems in discrete time, iteratively refining state predictions by leveraging posteriori state estimate $\hat{\boldsymbol{x}}$ and current measurements z_t while maintaining uncertainty through covariance matrices P. In object tracking applications, its process integrates state transition models F, Kalman gains K_t, observation models H, process noise Q, and observation noise R terms to alternately predict and update target states at each time step [7,8,10]:

$$\begin{cases} predict \begin{cases} \hat{\boldsymbol{x}}_{t|t-1} = F_t \hat{\boldsymbol{x}}_{t-1|t-1} \\ P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \end{cases} \\ update \begin{cases} K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t \right)^{-1} \\ \hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + K_t \left(z_t - H_t \hat{\boldsymbol{x}}_{t|t-1} \right) \\ P_{t|t} = (I - K_t H_t) P_{t|t-1} \end{cases} \end{cases} \tag{1}$$

The Kalman filter operates in two alternating phases: prediction, which estimates the next state using prior information from the current posterior, and update, which adjusts this estimate based on a new measurement derived via the observation model H.

We employ a standard Kalman filter with constant velocity motion and linear observation model, where the bounding box coordinates (x, y, a, h) serve as direct observations of the object state. The state vector $\boldsymbol{x}$ is defined as:

$$\boldsymbol{x} = \left[x, y, a, h, v_x, v_y, v_a, v_h \right]^T \tag{2}$$

where x, y represents the center coordinate of the bounding box, $a = w/h$ represents the aspect ratio of the target, h represents the height of the target, and the initial value is the position of the first frame of the target. v_x, v_y, v_a, v_h indicates the change rate of the corresponding variable with time and the initial value is 0.

For each mask M_i and its corresponding bounding box, $B_i = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ can be calculated by the following mathematical Formula (3):

$$\begin{cases} x_{\min} = \min\limits_{(i,j) \in S} j \\ x_{\max} = \max\limits_{(i,j) \in S} j \\ y_{\min} = \min\limits_{(i,j) \in S} i \\ y_{\max} = \max\limits_{(i,j) \in S} i \end{cases} \tag{3}$$

where $S = \left\{ (i,j) \middle| M_{i,j} > 0 \right\}$ represents the coordinate set of all non-zero (foreground) pixels in the mask.

Following [5], the KF-IoU fusion function is defined as follows:

$$\begin{aligned} iou_{weighted} &= \omega \cdot iou_{kf} + (1 - \omega) \cdot iou \\ iou_inds_{best} &= \text{argmax} \left(iou_{weighted} \right) \end{aligned} \tag{4}$$

where iou represents the intersection over union (IoU) of the original mask, iou_{kf} represents the IoU between the bounding box predicted by the Kalman filter and the actual detected bounding box (obtained by converting masks through Formula (3)). By calculating iou_{kf}, we can evaluate the accuracy of the Kalman filter prediction and combine it with the actual segmentation results to improve the stability and accuracy of tracking.

In multi-frame object tracking, the initialization and update process of the Kalman filter is as follows: First, when the mean and covariance of the Kalman filter are not initialized, or the number of stable frames is zero, the mask with the highest IoU is selected as the initial mask. The bounding box of this mask is then calculated to initialize the Kalman filter. Next, if the number of stable frames is less than the threshold, the boundary of the current frame is predicted first. The mask with the highest IoU is then selected, as choosing the best IoU mask ensures the accuracy of the predicted state. If the IoU of the selected mask is higher than the stability threshold, the mean and covariance matrix of the Kalman filter are updated, and the number of stable frames is incremented. Finally, when the number of stable frames exceeds the threshold and the system enters the stable state update process, the bounding box of the current frame is predicted. The bounding boxes of all candidate masks are calculated, and the mask with the highest weighted IoU (a combination of the IoU from the Kalman filter and the original IoU) is selected to update the mean and covariance matrices of the Kalman filter. The weighted IoU considers the confidence levels of different candidate objects, thereby selecting the best candidate. Before updating the state, it is determined whether the best IoU is below the set threshold. If it is, this indicates that the objects in the current frame do not match the predicted state, and the number of stable frames is reset.

In video sequence tracking, the KF-IoU fusion framework significantly reduces target mask jitter and enhances tracking stability across consecutive frames. By integrating Kalman filter predictions, the framework minimizes the influence of single-frame noise and assists in recovering target positions during partial occlusions. Weighted IoU improves robustness in low SNR environments where the quality of SAM2 model masks is unstable.

The KF-IoU fusion framework extends mask selection from a single frame to spatio-temporal joint optimization, significantly enhancing the stability and robustness of target tracking in consecutive frame scenarios. In contrast, the original SAM2 model selects the mask with the highest IoU without considering temporal continuity or prediction mechanisms, leading to potential instability and susceptibility to single-frame segmentation errors. By introducing the KF to predict the target motion trajectory and combining it with a dynamic weighted IoU decision-making strategy, spatio-temporal consistency in multi-mask output scenarios is significantly improved. Additionally, when the IoU is detected to be below the threshold, the stable frame count is reset to prevent error accumulation, further enhancing the robustness of the tracking system.

3.3. Adaptive Historical Frame Selection Strategy Based on Dynamic Threshold

The original SAM2 [2] selects historical frames at fixed time intervals (e.g., one frame per stride), which can introduce noise when frame quality is low and limits its ability to handle long-term occlusions or deformations. In this section, we propose an Adaptive Historical Frame Selection Strategy based on Dynamic Thresholds (AHFSS-DT) to improve the temporal coherence and robustness of frame selection.

Our method dynamically filters frames using a combination of IoU, object confidence scores, and motion prediction scores, ensuring selected frames maintain high semantic and spatial relevance. As shown in Figure 3, the strategy performs reverse traversal of historical frames to prioritize those that meet dynamically adjusted quality thresholds, while applying a fallback policy—including recent frame retention and gradual threshold

relaxation—when valid frames are insufficient. This design significantly enhances tracking continuity and segmentation accuracy in complex dynamic environments.

To address the limitations of fixed thresholds in complex scenarios, we designed a dynamic threshold generation function:

$$\begin{aligned} \tau_{IoU}^{(t)} &= \alpha \cdot \mathbb{E}[IoU_{hist}^{(t)}] + (1-\alpha) \cdot \tau_{IoU}^{\text{base}} \\ \tau_{obj}^{(t)} &= \beta \cdot \left(\max_{i<j} \text{ObjScore}_j - \text{ObjScore}_i \right) + \tau_{obj}^{\text{base}} \\ \tau_{KF}^{(t)} &= \gamma \cdot \frac{\varepsilon_{KF}}{\sigma_\varepsilon} + \tau_{KF}^{\text{base}} \end{aligned} \tag{5}$$

where α, β, γ are adaptive weight coefficients, $\mathbb{E}[\cdot]$ represents the statistical expectation of the IoU of historical frames, and τ_{base} is the preset baseline threshold. This function achieves dynamic adjustment of the IoU threshold τ_{IoU} and the object score threshold τ_{obj} by fusing historical statistical information with the feature differences of the current frame. For the motion prediction score τ_{KF}, we further introduce the Kalman filter prediction error ε_{KF} as a constraint, where σ_ε is the standard deviation of the error.

The validity of the historical frame f_i is determined by the following conditions simultaneously [5]:

$$\begin{cases} IoU(f_t, f_i) \geq \tau_{IoU}^{(t)} \\ ObjScore(f_i) \geq \tau_{obj}^{(t)} \\ KFConsistency(f_i) \geq \tau_{KF}^{(t)} \end{cases} \tag{6}$$

Through this joint constraint, it is ensured that the screened frames meet the dynamic standards in the three dimensions of spatial overlap degree, detection confidence, and motion consistency. When the number of valid frames $N_{valid} < N_{\max}$, the system starts the fallback policy: (1) recent frame forced retention, forcing the most recent frame f_{t-1} to be included in the candidate set; (2) gradient relaxation filling, reverse traversing historical frames and gradually reducing the threshold $\tau \leftarrow \delta \cdot \tau$, $\delta \in (0,1)$ until $N_{valid} = N_{\max}$. where $N_{\max}$ is the maximum number of frames to look back. This multi-dimensional quality assessment mechanism enables the model to dynamically filter noise frames and prioritize the retention of historical information with high semantic relevance and motion coherence.

The proposed AHFSS-DT enhances the cross-frame attention mechanism's ability to utilize temporal context efficiently. Compared to fixed-interval strategies, our method introduces minimal overhead in threshold computation ($O(D)$, where D is the feature dimension) and conditional checks $O(N_{hist})$, resulting in an overall complexity of $O(T + N_{hist})$. By constraining the encoder input sequence length $N_{\max}$, we avoid redundant computation while maintaining high performance in long-term tracking and occlusion recovery tasks.

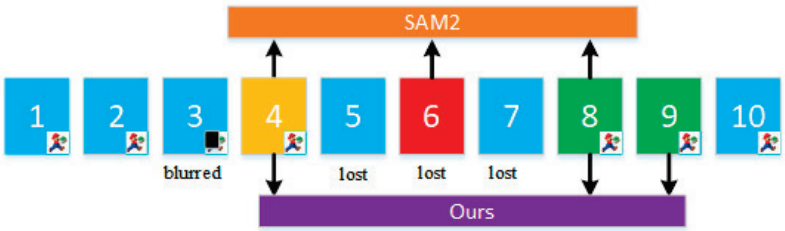


Figure 3. Adaptive historical frame selection. Green indicates highly correlated frames (IoU, object score, and KF score all meet the standards); yellow indicates partially valid frames (IoU meets the standard, but object score or KF score is insufficient); red indicates low-quality frames (IoU or content-independent).

3.4. Memory Management and Optimization for Efficient Video Segmentation

In this section, we present a comprehensive set of optimizations to manage memory efficiently during video segmentation tasks using the SAM2. These optimizations aim to maintain constant VRAM and RAM usage, even for infinitely long video sequences, while preserving the accuracy and efficiency of the original model.

3.4.1. Dynamic Memory Management

In processing long video sequences, memory growth can become unbounded if outdated frames and intermediate results are not effectively released. We address this issue through the following mechanisms:

- Offloading frames to CPU RAM: By setting the offload_video_to_cpu parameter in the init_state method [39], video frames are transferred from GPU VRAM to CPU RAM, reducing the GPU memory footprint without sacrificing availability during inference.
- Manual VRAM release after memory attention calculation: Intermediate variables generated during the Memory Attention computation were found to persist unnecessarily in VRAM. We added explicit cleanup operations to release these variables immediately after use, significantly lowering peak memory usage.

3.4.2. Efficient Propagation and Result Management

To ensure prompt correction across video frames while minimizing computational overhead, we apply the following propagation and result management techniques:

- Limiting propagation length: We cap the propagation length at M, ensuring that each correction operation only affects a fixed number of historical frames. This balances correction robustness with computational cost, reducing the total number of frames processed during inference from N to approximately(M/K) * N, where K is the frame buffer size.
- Immediate release of processed segmentation results: We observed that caching segmentation results in the video_segments dictionary led to linear memory growth. To counteract this, we implemented proactive deallocation, releasing each frame's segmentation results immediately after use. This prevents unbounded dictionary expansion and stabilizes memory usage.

3.4.3. Continuous Old Frame Clearing

As shown in Figure 4, we introduce a dynamic mechanism to continuously evict outdated frame data during inference on long or infinite video sequences. This is governed by the max_inference_state_frames parameter, which determines the number of most recent frames to retain in memory.

- Retention Threshold Mechanism: Only frames beyond the max_inference_state_frames threshold from the current frame are cleared. Crucially, max_inference_state_frames [39,40] must be greater than max_frame_num_to_track in the propagate_in_video() function to ensure that historically relevant frames are retained until no longer needed.
- Automatic Cleanup After Propagation: Following each propagation step, frames exceeding the retention limit are automatically released from memory. Frames stored in the preload memory bank are preserved indefinitely.
- Memory Bounds During Inference: The upper bound of VRAM usage is dictated by the memory required for newly processed frames, retained frames, and the fixed-size preload memory bank. The lower bound corresponds to the memory required for the maximum propagation length plus the preload bank. This strategy ensures memory usage remains within a predictable range, effectively balancing efficiency and safety.

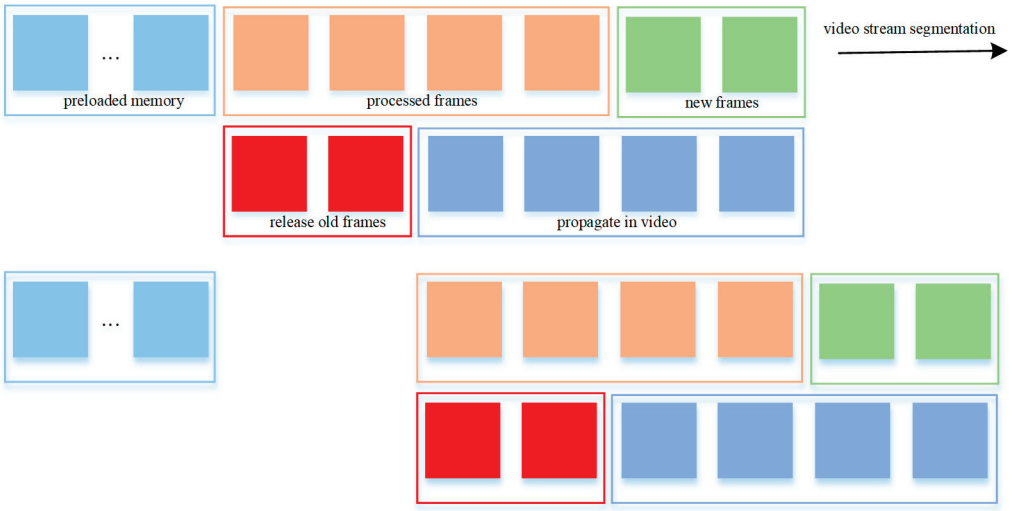


Figure 4. Continuous old frame clearing process.

3.4.4. Advanced Memory Optimization Techniques

To further reduce linear memory growth and maintain constant VRAM consumption during prolonged inference, we employ the following enhancements:

- Index Mapping Decoupling: We refactored frame access logic to use an independent frame index mapping table, decoupling logical frame references from physical storage. This allows precise tracking of required frames without relying on sequential indexing.
- Dynamic State Synchronization: The frame index mapping table is dynamically maintained during state initialization and updates, ensuring real-time alignment between logical indices and the current frame sequence.
- Coordinated Cleanup Strategy: A dedicated function systematically removes outdated frames that exceed retention thresholds. This function synchronizes updates to both the image cache container and the logical index table, preserving data-reference consistency.
- Global Frame Statistics: We modified the historical frame counter to track the total number of frames ever loaded rather than just those currently active. This distinction enables accurate memory accounting and avoids overestimating active storage needs.

These steps ensure that VRAM usage remains constant, as outdated frames are immediately released after their relevance expires, while the preload memory bank and propagated frames are retained within bounded limits. The system avoids linear memory growth by dynamically managing frame access and deletion through the index mapping and state synchronization mechanisms.

By implementing these optimizations, we achieve a robust and efficient memory management system for the SAM2 model. This system ensures that the VRAM and RAM usage remain constant, even when processing infinitely long video sequences. The pipeline maintains the same efficiency and accuracy as the original SAM2, while reducing performance overhead through a series of engineering enhancements. These optimizations are particularly useful for applications with limited GPU memory and for processing long video sequences.

4. Experiments

4.1. Metrics and Datasets

To assess our method, we utilize two widely-adopted VOS benchmarks [2,14] and use evaluation metrics: region similarity J (IoU-based), contour accuracy F (harmonic mean of precision and recall), and their arithmetic mean $J\&F$ as an overall performance

indicator [3]. Experiments are performed in a semi-supervised framework where the initial frame's mask is provided. Specifically, region similarity is calculated as the IoU between predicted and ground-truth masks, while contour accuracy evaluates boundary alignment via the harmonic mean of precision and recall. The combined metric averages these values to summarize system efficacy, where $\hat{M}, M$ are the predicted segmentation masks and the ground truth masks, respectively.

$$J = \frac{\hat{M} \cap M}{\hat{M} \cup M} \tag{7}$$

$$F = \frac{2P_c R_c}{P_c + R_c} \tag{8}$$

$$J\&F = \frac{J + F}{2} \tag{9}$$

We will use the following two VOS datasets to evaluate our method:

SA-V (Segment Anything in Video) [2] is a large-scale video segmentation benchmark specifically designed for promptable visual analysis across diverse scenarios, comprising 50.9 K video clips with 35.5 million high-precision masks and 642.6 K localized masklets (short-term object instances). This dataset emphasizes challenging real-world segmentation tasks, including persistent occlusions, small objects (occupying < 5% of frame area), and long-term reappearance of objects under varying appearances. To ensure comprehensive model evaluation, SA-V is partitioned into 45,495 training videos (89.4% of total data) for robust feature learning, 155 validation videos (293 masklets) for hyperparameter optimization, and 150 testing videos (278 masklets) for rigorous generalization assessment. By prioritizing spatio-temporal consistency and minimizing data leakage, SA-V establishes a critical benchmark for advancing segmentation robustness in dynamic environments, particularly in handling discontinuous object trajectories and cross-frame ambiguities.

The LVOS v1 [22] benchmark pioneers long-term video object segmentation in practical scenarios, comprising 720 video clips (296,401 frames, 407,945 annotations) with an average duration exceeding 60 s. It introduces critical challenges such as long-term object reoccurrence and cross-temporal interference from similar objects, partitioned into 120 training, 50 validation, and 50 testing videos. Building upon this foundation, LVOS v2 [22] extends the dataset to 420 training, 140 validation, and 160 testing videos, fully incorporating all sequences from LVOS v1 while adding new sequences for enhanced complexity. Covering 44 categories of real-world scenarios, LVOS v2 deliberately excludes 12 categories from the training set to rigorously evaluate the generalization capability of VOS models, particularly in handling unseen object types. This version serves as the primary benchmark in our work due to its expanded scale, comprehensive coverage of LVOS v1 data, and emphasis on real-world robustness evaluation.

4.2. Environment and Parameters

All experiments are performed on an NVIDIA A40 GPU with 48 GB of memory, using Python 3.10, PyTorch 2.5.1, CUDA 12.4, and Ubuntu 22.04. The Kalman filter is initialized using the bounding box associated with the best segmentation mask, and stable tracking is achieved after 15 consecutive frames. The tracking system uses the following hyperparameters: a stable IoU threshold of 0.3, a minimum object score logit of 0.3, and a Kalman filter confidence weight of 0.25. For the memory bank, the IoU threshold is set to 0.5, and both the object score and Kalman filter score thresholds are set to 0.0. All remaining parameters are inherited from the default settings in SAM2.

4.3. Quantitative Results

The quantitative results presented in Tables 1–3 provide a comprehensive comparison of the performance of our proposed method (SAM2Plus) against SAM2 and SAM2Long across different datasets and model sizes.

SA-V Datasets (Table 1): The SA-V dataset consists of short videos (approximately 10 s in length). In these scenarios, the performance improvements of our method, SAM2Plus, are less pronounced compared to SAM2 and SAM2Long. For example, in the SA-V Val (Large) set, SAM2.1Plus achieves a score of 79.6 (J&F), which is slightly better than SAM2.1 (78.6) but not as significant as the improvements seen in longer videos. This is because short videos have fewer frames, and the benefits of our adaptive historical frame selection strategy based on dynamic thresholds (AHFSS-DT) are less evident in such scenarios. However, despite the smaller improvement in J&F scores, SAM2Plus maintains a similar memory usage and inference speed compared to SAM2, making it more practical for real-time applications.

LVOS v2 Datasets (Table 2): The LVOS v2 dataset consists of longer videos, and our method, SAM2Plus, demonstrates more significant performance improvements. The adaptive historical frame selection strategy based on dynamic thresholds (AHFSS-DT) and the integration of KF to predict the target's motion trajectory significantly enhance the efficiency of cross-frame attention mechanisms, leading to better adaptability to dynamic and complex scenes. For instance, in the LVOS v2 Val (Base+) set, SAM2.1Plus achieves a score of 86.3 (J&F), compared to 83.1 for SAM2 and 85.2 for SAM2Long. While SAM2Long shows a higher J&F score in the LVOS v2 Val (Tiny, Small) set, it comes at a significant cost in terms of memory usage and inference speed, as shown in Table 3.

Table 1. Performance comparison of SA-V datasets (SAM2, SAM2Long, and SAM2Plus) across all model sizes.

Method	Backbone	SA-V Val			SA-V Test		
		J&F	*J*	*F*	*J&F*	*J*	*F*
SAM2	Tiny	73.5	70.1	76.9	74.6	71.1	78.0
SAM2Long		77.0	73.2	80.7	78.7	74.6	82.7
SAM2Plus		74.3	70.4	78.1	76.4	72.4	80.3
SAM2.1		75.1	71.6	78.6	76.3	72.7	79.8
SAM2.1Long		78.9	75.2	82.7	79.0	75.2	82.9
SAM2.1Plus		75.5	71.7	79.4	76.7	72.8	80.7
SAM2	Small	73.0	69.7	76.3	74.6	71.0	78.1
SAM2Long		77.7	73.9	81.5	78.1	74.1	82.0
SAM2Plus		74.3	70.3	78.2	75.4	71.4	79.4
SAM2.1		76.9	73.5	80.3	76.9	73.3	80.5
SAM2.1Long		79.6	75.9	83.3	80.4	76.6	84.1
SAM2.1Plus		77.3	73.5	81.1	77.6	73.8	81.5
SAM2	Base+	75.4	71.9	78.8	74.6	71.2	78.1
SAM2Long		78.4	74.7	82.1	78.5	74.7	82.2
SAM2Plus		75.4	71.6	79.2	77.1	73.1	81.1
SAM2.1		78.0	74.6	81.5	77.7	74.2	81.2
SAM2.1Long		80.5	76.8	84.2	80.8	77.1	84.5
SAM2.1Plus		77.4	73.6	81.2	78.3	74.5	82.1
SAM2	Large	76.3	73.0	79.5	75.5	72.2	78.9
SAM2Long		80.8	77.1	84.5	80.8	76.8	84.7
SAM2Plus		77.4	73.6	81.2	79.0	74.9	83.0
SAM2.1		78.6	75.1	82.0	79.6	76.1	83.2
SAM2.1Long		81.1	77.5	84.7	81.2	77.6	84.9
SAM2.1Plus		79.6	75.8	83.4	79.6	75.8	83.4

Table 2. Performance comparison of LVOS v2 datasets (SAM2, SAM2Long, and SAM2Plus) across all model sizes.

Method	Backbone	LVOS v2 Train			LVOS v2 Val		
		J&F	*J*	*F*	*J&F*	*J*	*F*
SAM2	Tiny	94.5	92.9	96.1	77.8	74.5	81.2
SAM2Long		95.2	93.6	96.9	81.4	77.7	85.0
SAM2Plus		95.2	93.6	96.9	80.0	76.3	83.6
SAM2.1		95.0	93.4	96.7	81.6	77.9	85.2
SAM2.1Long		95.2	93.6	96.9	82.4	78.8	85.9
SAM2.1Plus		95.1	93.5	96.8	83.5	79.7	87.3
SAM2	Small	94.6	92.9	96.2	79.7	76.2	83.3
SAM2Long		95.2	93.5	96.9	83.2	79.5	86.8
SAM2Plus		95.1	93.5	96.8	81.4	77.7	85.2
SAM2.1		95.4	93.7	97.0	82.1	78.6	85.6
SAM2.1Long		95.6	94.0	97.3	84.3	80.7	88.0
SAM2.1 Plus		95.4	93.7	97.0	84.8	81.0	88.5
SAM2	Base+	95.3	93.7	96.9	80.2	76.8	83.6
SAM2Long		95.5	93.9	97.2	82.3	78.8	85.9
SAM2 Plus		95.5	93.9	97.2	82.9	79.3	86.5
SAM2.1		95.6	94.0	97.2	83.1	79.6	86.5
SAM2.1Long		95.6	94.0	97.2	85.2	81.5	88.9
SAM2.1Plus		95.6	94.0	97.2	86.3	82.6	90.0
SAM2	Large	95.2	93.6	96.8	83.0	79.6	86.4
SAM2Long		95.7	94.1	97.3	85.2	81.8	88.7
SAM2Plus		95.7	94.1	97.4	85.5	81.9	89.1
SAM2.1		95.3	93.7	96.9	84.0	80.7	87.4
SAM2.1Long		95.7	94.0	97.3	85.3	81.9	88.8
SAM2.1Plus		95.8	94.1	97.4	85.6	82.0	89.3

Table 3. Performance comparison of 1080p videos in the LVOS v2 dataset based on A40 inference: SAM2, SAM2Long and SAM2Plus of all model sizes.

Method	Backbone	GPU Memory (MB)	Speed (FPS)
SAM2	Tiny	4165	30.87
SAM2Long		5707	13.99
SAM2Plus		4171	28.38
SAM2.1		4165	30.21
SAM2.1Long		5707	12.81
SAM2.1Plus		4173	28.16
SAM2	Small	4195	26.87
SAM2Long		5763	13.32
SAM2Plus		4205	26.63
SAM2.1		4193	26.65
SAM2.1Long		5811	12.81
SAM2.1Plus		4203	26.33
SAM2	Base+	4447	21.32
SAM2Long		5971	11.69
SAM2Plus		4453	20.35
SAM2.1		4449	20.11
SAM2.1Long		5971	11.71
SAM2.1Plus		4453	20.26
SAM2	Large	5497	13.69
SAM2Long		6779	8.47
SAM2Plus		5505	13.55
SAM2.1		5499	13.13
SAM2.1Long		6913	8.51
SAM2.1Plus		5503	13.46

The evaluation of 1080p videos from the LVOS v2 dataset under A40 GPU settings demonstrates that SAM2Plus achieves a well-balanced trade-off between memory usage and inference speed. Compared to the original SAM2, SAM2Plus slightly increases GPU memory consumption (e.g., 5503 MB vs. 5499 MB with the Large backbone), but maintains comparable inference speed (13.46 FPS vs. 13.13 FPS). This indicates that the additional modules introduced in SAM2Plus (e.g., the KF-IoU fusion and enhanced memory attention) have a minimal impact on frame rate, while contributing to improved tracking consistency. In contrast, SAM2Long, which extends the memory window further to model longer temporal contexts, achieves higher segmentation accuracy (J&F scores) in some cases, but at the cost of significantly increased memory usage (up to 6913 MB) and reduced inference speed (8.51 FPS). This trade-off makes it less suitable for real-time or extended-duration video processing. Therefore, SAM2Plus is considered a more practical and balanced solution for real-time applications, provided that sufficient GPU memory is available. It delivers enhanced performance over the baseline without sacrificing frame rate, and is particularly effective in production environments where moderate resource allocation and high tracking accuracy are both prioritized.

State-of-the-Art Comparison (Table 4): Table 4 presents a performance comparison of various methods on the LVOS v2 validation set. From the table, it is evident that SAM2.1Plus performs the best overall, achieving a J&F score of 86.3. It excels in both seen and unseen categories, with J scores of 81.6 and 83.3, and F scores of 89.0 and 91.2, respectively. Although SAM2.1Long scores slightly higher in some categories, particularly in the unseen category, its significantly increased resource consumption makes it less suitable for real-time applications. In contrast, SAM2.1Plus not only approaches the highest performance but also demonstrates excellent resource efficiency, making it the best choice for real-time applications and long video analysis in production environments. Other methods, such as SAM2Plus and SAM2.1, also show good performance but are slightly inferior to SAM2.1Plus. Overall, SAM2.1Plus achieves the best balance between performance and resource efficiency.

Table 4. Cross-architecture benchmarking of LVOS validation split, employing subscript notations s/u to distinguish metric distributions between known and novel semantic taxonomies under standardized evaluation frameworks [41].

Method	LVOS v2 Val				
	J&F	J_s	F_s	J_u	F_u
LWL [42]	60.6	58.0	64.3	57.2	62.9
CFBI [16]	55.0	52.9	59.2	51.7	56.2
STCN [43]	60.6	57.2	64.0	57.5	63.8
RDE [44]	62.2	56.7	64.1	60.8	67.2
DeAOT [45]	63.9	61.5	69.0	58.4	66.6
XMem [46]	64.5	62.6	69.1	60.6	65.6
SAM 2 [2]	79.8	80.0	86.6	71.6	81.1
SAM 2.1 [2]	84.1	80.7	87.4	80.6	87.7
SAM2Long [4]	84.2	82.3	89.2	79.1	86.2
SAM2.1Long [4]	85.9	81.7	88.6	83.0	90.5
SAM2Plus	84.8	82.2	89.4	80.3	87.3
SAM2.1Plus	86.3	81.6	89.0	83.3	91.2

These results highlight the robustness and adaptability of our proposed method, SAM2.1Plus, particularly in long video tracking and occlusion recovery tasks. The improvements in performance on the LVOS v2 dataset, which consists of longer videos, underscore the effectiveness of our method in handling dynamic and complex scenes. In contrast, the performance gains on the SA-V dataset, which consists of shorter videos, are less pronounced,

but still demonstrate the overall improvement in tracking continuity and segmentation quality. Additionally, the efficient memory usage and inference speed of SAM2.1Plus make it a more practical choice for real-world applications compared to SAM2Long.

4.4. Qualitative Results

As illustrated in Figure 5, the visualization results highlight the performance differences between our method, SAM2, and SAM2Long in various scenarios. When the target of interest is occluded or temporarily disappears and then re-enters the region of interest, SAM2 fails to maintain the target's tracking. Even upon re-emergence, SAM2 is unable to resume tracking. In contrast, our method can immediately resume tracking the target as soon as it reappears after short-term occlusions or disappearances. This capability is attributed to the integration of KF and multi-frame tracking strategies, which enable accurate position prediction and rapid tracking recovery.

In scenarios where the target moves rapidly, SAM2 exhibits lower segmentation quality and is prone to target loss or inaccurate segmentation. SAM2Long shows some improvements but still suffers from suboptimal segmentation quality in such scenarios. Our method, however, achieves higher segmentation quality in rapid movement scenarios. By introducing KF to predict the target's motion trajectory and combining it with a dynamic weighted IoU decision strategy, our method effectively mitigates the impact of single-frame noise, ensuring consistent and accurate tracking.

In complex dynamic environments, SAM2's tracking and segmentation performance often deteriorates due to background interference and target occlusions. While SAM2Long offers some improvement, it still struggles in such scenarios. To address these limitations, we propose an AHFSS-DT for video object tracking. This strategy enhances the cross-frame attention mechanism's ability to exploit temporal context by incorporating dynamic thresholds, multi-metric evaluation, and flexible frame selection. As a result, our method achieves improved adaptability to dynamic scenes and better performance in long video tracking and occlusion recovery. The visualizations in Figure 5 further demonstrate its superior tracking continuity and segmentation quality in challenging scenarios involving rapid motion, occlusion, and target reappearance—highlighting its robustness and practical potential.

4.5. Long Video Segmentation in Real Scenes with SAM2Plus

To evaluate the performance of our proposed SAM2Plus in long video segmentation tasks, we conduct all experiments on a real-world 4K-resolution video stream captured from typical urban environments, including parks and roadways. The video has a frame rate of 25 fps and a total duration of approximately 5–10 min. For computational efficiency, we down-sample the video to 8 fps, which strikes a balance between processing speed and temporal resolution.

All experiments are performed on an NVIDIA A40 GPU with 48 GB of memory, using Python 3.10, PyTorch 2.5.1, CUDA 12.4, and Ubuntu 22.04. We utilize the YOLOv11 detection model [47], introduced by Ultralytics [26] in 2024, to generate object bounding boxes in the first frame. YOLOv11 builds upon the architecture of YOLOv8 [27] with significant improvements in both model design and training methodology, making it well-suited for industrial deployment. In our setup, the detected boxes correspond to human subjects, non-motorized vehicles, and motorized vehicles, which are then converted into masks and used as initial prompts for SAM2.1-Base+. Thanks to the robust correction mechanism in SAM2Plus, any newly introduced prompts can be retroactively applied to all previously inferred frames, enabling iterative refinement and reducing mask drift over time. This allows us to achieve effective object tracking and segmentation throughout the video sequence using only a single frame of initial annotations.

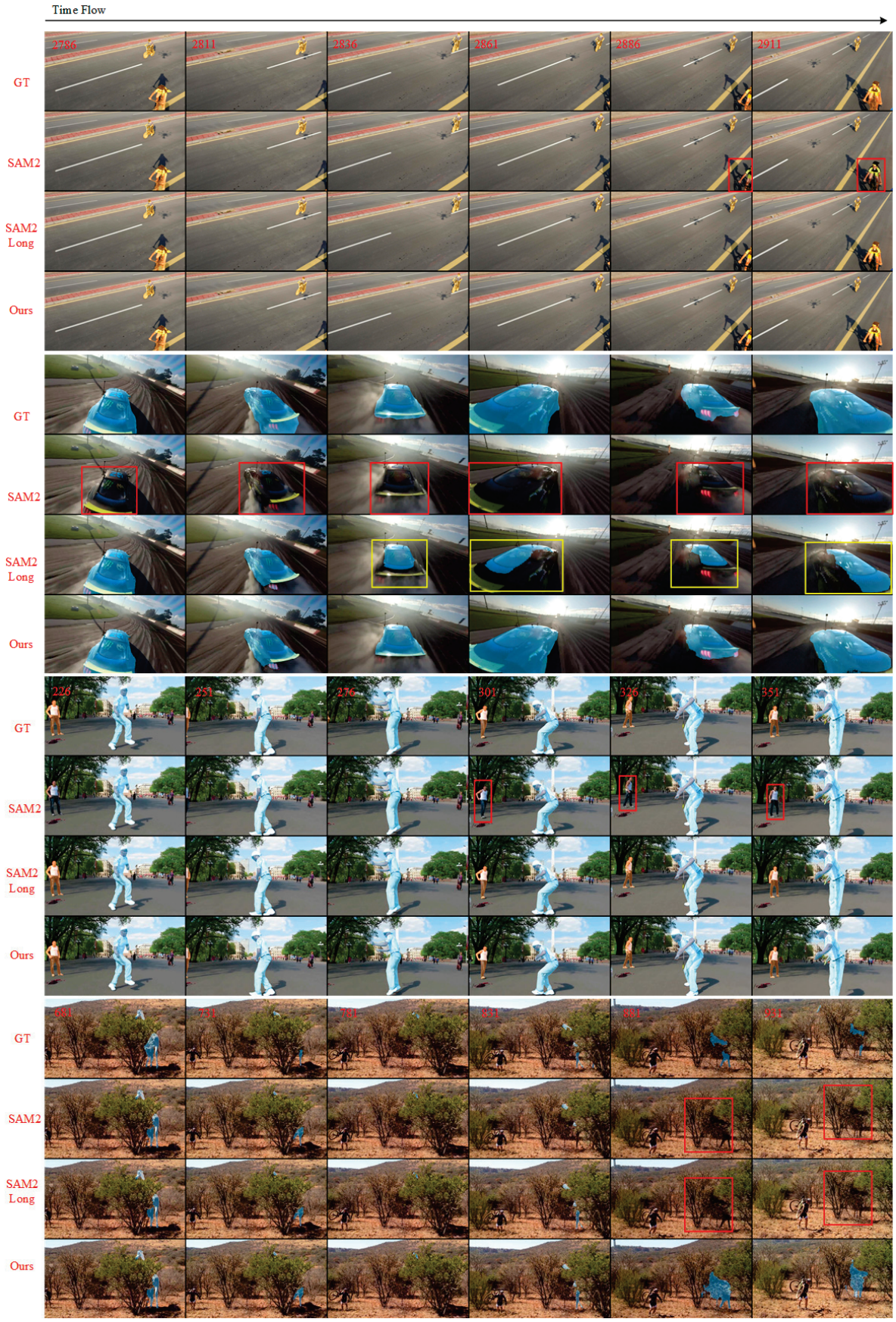


Figure 5. Qualitative comparison of SAM2, SAM2Long and our method, with GT (ground truth) as the reference. The red boxes represent the missing objects and the yellow boxes are used to highlight the low-quality segmentation results.

From Figure 6, it is evident that, in dense target scenes, the object masks obtained from the detection boxes in the first frame can overlap. However, SAM2Plus can correct these mask deviations within about 5 frames, resulting in accurate and stable tracking of the true object masks. Additionally, the system can effectively re-track targets that experience short-term occlusions or reappear after disappearing. This capability further demonstrates the robustness of our method in handling long-term object segmentation and tracking. For the target segmentation of one video channel, the video memory of the SAM2.1-Base+ model stabilized at 1550 MB, indicating efficient memory usage.

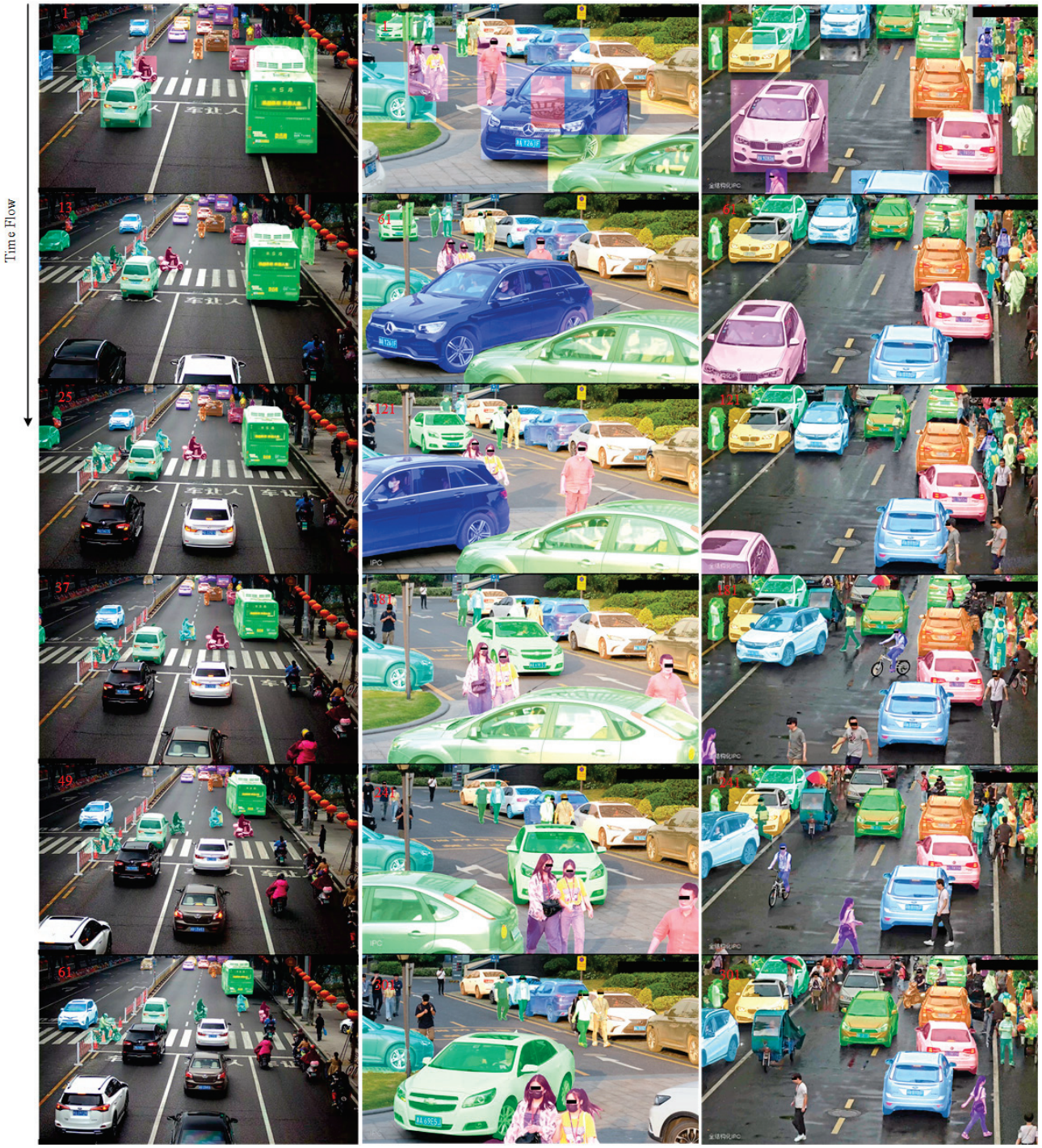


Figure 6. Long video segmentation and tracking results using SAM2Plus based on the detection box in the first frame. The red numbers in the picture represent the video frame numbers.

The combination of YOLOv11 for initial object detection and SAM2.1-Base+ for segmentation and tracking proved to be highly effective for low-speed scene analysis. The system achieved high accuracy, efficient processing, and robust error correction, making it suitable for industrial deployment in various applications such as smart parks and traffic management.

5. Discussion

We conducted a series of ablation experiments on the LVOS v2 validation set using SAM2-Base+ as the default model size to observe the influence of each module on the overall segmentation results. It can be seen from Table 5 that both of these two modules have had a positive impact on the SAM2 model. The KF-IoU fusion framework increased by approximately 1%, and the AHFSS-DT module increased by 1.7%. The combination of the two on the LVOS v2 dataset can achieve the best *J*&*F*, which are 82.9% and 86.3 respectively. The results of the above ablation experiments illustrate that the KF-IoU module significantly reduces the target mask jitter and enhances the tracking stability across consecutive frames. Meanwhile, the AHFSS-DT module significantly improves the efficiency of the cross-frame attention mechanism. The adaptability to dynamic and complex scenes has been improved, and better performance has been achieved in long video tracking and occlusion recovery tasks.

Table 5. Ablation on the effectiveness of the proposed modules. Performance is evaluated on 1080p videos from the LVOS v2 dataset using A40 GPU inference, with GPU memory usage and FPS reported as efficiency metrics.

Method	KF-IoU	AHFSS-DT	LVOS v2 Val			GPU Memory (MB)	Speed (FPS)
			J&*F*	*J*	*F*		
SAM2			80.2	76.8	83.6	4447	21.32
	√		81.2	77.5	84.8	4450	20.86
	√	√	82.9	79.3	86.5	4453	20.35
SAM2.1			83.1	79.6	86.5	4449	20.11
	√		84.5	80.8	88.2	4452	20.19
	√	√	86.3	82.6	90.0	4453	20.26

6. Conclusions

In this work, we present SAM2Plus, a zero-shot enhancement framework designed to overcome the long-term segmentation limitations of Segment Anything Model 2 (SAM 2) in video object segmentation (VOS). By unifying physics-based motion prediction via the Kalman filter, dynamic quality thresholds, and a novel adaptive memory management system, SAM2Plus achieves significant improvements in segmentation accuracy and reliability across extended video sequences. First, the Kalman filter component models object motion dynamics using physical constraints to predict trajectories and dynamically refine segmentation states, effectively mitigating positional drift during occlusions or abrupt velocity changes. Second, prioritizing high-quality frames and balancing confidence scores with temporal smoothness, dynamic thresholds are coupled with multi-criteria metrics (e.g., motion coherence, appearance consistency), which reduce ambiguities in complex scenes such as crowded environments or those involving fast-moving objects. Then, the optimized memory architecture also ensures computational efficiency by dynamically pruning outdated or low-confidence segmentation records while preserving temporally coherent spatial context, thereby maintaining constant memory usage even for infinitely long videos. Finally, extensive experiments on two VOS benchmarks confirm that SAM2Plus

significantly outperforms SAM 2, achieving an average 1.0 improvement in J&F metrics across 24 comparisons and exceeding it by over 2.3 points on SA-V and LVOS datasets for long-term VOS, while enabling real-time operation without additional parameters or fine-tuning and enhancing adaptability to dynamic environments. In the future, we will further optimize the target occlusion problem and explore the semantic interaction among multiple targets within the same framework. Overall, SAM2Plus represents a significant advancement in VOS, offering a powerful tool for a wide range of applications in both images and videos, particularly in dynamic and complex environments.

Author Contributions: Conceptualization, H.S. and J.Y.; methodology, F.W. and P.H.; software, H.S. and P.H.; validation, H.S., J.Y., Y.Q. and F.W.; formal analysis, H.S. and Y.Q.; investigation, J.Y. and P.H.; resources, H.S.; data curation, F.W. and Y.Q.; writing—original draft preparation, F.W.; writing—review and editing, H.S. and Y.Q.; visualization, H.S.; supervision, J.Y. and P.H.; project administration, J.Y.; funding acquisition, J.Y. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All relevant data presented in the article are stored according to institutional requirements and, as such, are not available online. However, all data used in this manuscript can be made available upon request to the authors.

Acknowledgments: We sincerely thank Zhejiang Dahua Technology Co., Ltd. for providing product data support for the verification experiment. Furthermore, we would like to express our particular gratitude to Huang Peng for his support and assistance in this project.

Conflicts of Interest: Authors Hao Su, Peng Huang and Yuetong Qixuan were employed by the Zhejiang Dahua Technology Co., Ltd., company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 4015–4026.
2. Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.K.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. *arXiv* **2024**, arXiv:2408.00714.
3. Jiaxing, Z.; Hao, T. SAM2 for Image and Video Segmentation: A Comprehensive Survey. *arXiv* **2025**, arXiv:2503.12781.
4. Ding, S.; Qian, R.; Dong, X.-W.; Zhang, P.; Zang, Y.; Cao, Y.; Guo, Y.; Lin, D.; Wang, J. SAM2Long: Enhancing SAM 2 for Long Video Segmentation with a Training-Free Memory Tree. *arXiv* **2024**, arXiv:2410.16268.
5. Yang, C.-Y.; Huang, H.-W.; Chai, W.; Jiang, Z.; Hwang, J.-N. SAMURAI: Adapting Segment Anything Model for Zero-Shot Visual Tracking with Motion-Aware Memory. *arXiv* **2024**, arXiv:2411.11922.
6. Basar, T. Contributions to the Theory of Optimal Control. 2001; pp. 147–166. Available online: https://www.jstor.org/stable/1993703 (accessed on 2 July 2025).
7. Cao, J.; Weng, X.; Khirodkar, R.; Pang, J.; Kitani, K. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2022; pp. 9686–9696.
8. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.

9. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *Computer Vision—ECCV 2022, Proceedings of the 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Springer: Berlin/Heidelberg, Germany, 2022.
10. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple Online and Realtime Tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
11. Yang, S.; Thormann, K.; Baum, M. Linear-Time Joint Probabilistic Data Association for Multiple Extended Object Tracking. In Proceedings of the 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), Sheffield, UK, 8–11 July 2018; pp. 6–10.
12. Blackman, S.S. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerosp. Electron. Syst. Mag.* **2004**, *19*, 5–18. [CrossRef]
13. Chaoqun, Y.; Xiaowei, L.; Zhiguo, S.; Heng, Z.; Xianghui, C. Augmented LRFS-based filter: Holistic tracking of group objects. *Signal Process.* **2025**, *226*, 109665. [CrossRef]
14. Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; Gool, L.V. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv* **2017**, arXiv:1704.00675.
15. Oh, S.W.; Lee, J.-Y.; Xu, N.; Kim, S.J. Space-Time Memory Networks for Video Object Segmentation with User Guidance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 442–455. [CrossRef] [PubMed]
16. Yang, Z.; Wei, Y.; Yang, Y. Collaborative Video Object Segmentation by Foreground-Background Integration. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; pp. 332–348.
17. Qin, Z.; Zhou, S.; Wang, L.; Duan, J.; Hua, G.; Tang, W. MotionTrack: Learning Robust Short-Term and Long-Term Motions for Multi-Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 17939–17948.
18. Caelles, S.; Maninis, K.K.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; Van Gool, L. One-Shot Video Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 5320–5329.
19. Chou, Y.S.; Wang, C.Y.; Lin, S.D.; Liao, H.Y.M. How Incompletely Segmented Information Affects Multi-Object Tracking and Segmentation (MOTS). In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 2086–2090.
20. Chen, Z.; Fang, G.; Ma, X.; Wang, X. SlimSAM: 0.1\% Data Makes Segment Anything Slim. In Proceedings of the Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024; pp. 39434–39461.
21. Cuttano, C.; Trivigno, G.; Rosi, G.; Masone, C.; Averta, G. SAMWISE: Infusing wisdom in SAM2 for Text-Driven Video Segmentation. *arXiv* **2024**, arXiv:2411.17646.
22. Hong, L.; Liu, Z.; Chen, W.; Tan, C.; Feng, Y.; Zhou, X.; Guo, P.; Li, J.; Chen, Z.; Gao, S.; et al. LVOS: A Benchmark for Large-scale Long-term Video Object Segmentation. *arXiv* **2024**, arXiv:2404.19326.
23. Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; Huang, T. YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; Proceedings, Part V. pp. 603–619.
24. Ding, H.; Liu, C.; He, S.; Jiang, X.; Torr, P.H.S.; Bai, S. MOSE: A New Dataset for Video Object Segmentation in Complex Scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 20167–20177.
25. Zhao, X.; Ding, W.-Y.; An, Y.; Du, Y.; Yu, T.; Li, M.; Tang, M.; Wang, J. Fast Segment Anything. *arXiv* **2023**, arXiv:2306.12156.
26. Ultralytics. Available online: https://github.com/ultralytics/ultralytics (accessed on 6 May 2025).
27. Varghese, R.; Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 17–18 April 2024.
28. Zhang, C.; Han, D.; Qiao, Y.; Kim, J.U.; Bae, S.-H.; Lee, S.; Hong, C.-S. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. *arXiv* **2023**, arXiv:2306.14289.
29. Zhang, C.; Han, D.; Zheng, S.; Choi, J.H.; Kim, T.-H.; Hong, C.-S. MobileSAMv2: Faster Segment Anything to Everything. *arXiv* **2023**, arXiv:2312.09579.
30. Xiong, Y.; Varadarajan, B.; Wu, L.; Xiang, X.; Xiao, F.; Zhu, C.; Dai, X.; Wang, D.; Sun, F.; Iandola, F.; et al. EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024.
31. Song, Y.; Pu, B.; Wang, P.; Jiang, H.; Dong, D.; Cao, Y.; Shen, Y. SAM-Lightening: A Lightweight Segment Anything Model with Dilated Flash Attention to Achieve 30 times Acceleration. *arXiv* **2024**, arXiv:2403.09195.

32. Zhang, Z.; Cai, H.; Han, S. EfficientViT-SAM: Accelerated Segment Anything Model Without Performance Loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 17–21 June 2024; pp. 7859–7863.
33. Zhou, C.; Li, X.; Loy, C.C.; Dai, B. EdgeSAM: Prompt-In-the-Loop Distillation for On-Device Deployment of SAM. *arXiv* **2023**, arXiv:2312.06660.
34. Ma, J.; He, Y.; Li, F.; Han, L.; You, C.; Wang, B. Segment anything in medical images. *Nat. Commun.* **2024**, *15*, 654. [CrossRef] [PubMed]
35. Chen, T.; Lu, A.; Zhu, L.; Ding, C.; Yu, C.; Ji, D.; Li, Z.; Sun, L.; Mao, P.; Zang, Y. SAM2-Adapter: Evaluating & Adapting Segment Anything 2 in Downstream Tasks: Camouflage, Shadow, Medical Image Segmentation, and More. *arXiv* **2024**, arXiv:2408.04579.
36. Videnovic, J.; Lukezic, A.; Kristan, M. A Distractor-Aware Memory for Visual Object Tracking with SAM2. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), Nashville, TN, USA, 11–15 June 2025.
37. Jiang, J.; Wang, Z.; Zhao, M.; Li, Y.; Jiang, D.S. SAM2MOT: A Novel Paradigm of Multi-Object Tracking by Segmentation. *arXiv* **2025**, arXiv:2504.04519.
38. Ryali, C.; Hu, Y.-T.; Bolya, D.; Wei, C.; Fan, H.; Huang, P.-Y.; Aggarwal, V.; Chowdhury, A.; Poursaeed, O.; Hoffman, J.; et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; Volume 202, pp. 29441–29454.
39. sam2. Available online: https://github.com/facebookresearch/sam2 (accessed on 6 May 2025).
40. Wang, Z.; Zhou, Q.; Liu, Z. Det-SAM2: Technical Report on the Self-Prompting Segmentation Framework Based on Segment Anything Model 2. *arXiv* **2024**, arXiv:2411.18977.
41. lvos-Evaluation. Available online: https://github.com/LingyiHongfd/lvos-evaluation (accessed on 6 May 2025).
42. Bhat, G.; Lawin, F.J.; Danelljan, M.; Robinson, A.; Felsberg, M.; Van Gool, L.; Timofte, R. Learning What to Learn for Video Object Segmentation. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
43. Cheng, H.K.; Tai, Y.W.; Tang, C.K. Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11781–11794.
44. Li, M.; Hu, L.; Xiong, Z.; Zhang, B.; Pan, P.; Liu, D. Recurrent Dynamic Embedding for Video Object Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
45. Yang, Z.; Wei, Y.; Yang, Y. Associating Objects with Transformers for Video Object Segmentation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 2491–2502.
46. Cheng, H.K.; Schwing, A.G. XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model. In *Computer Vision—ECCV 2022, Proceedings of the 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Springer: Cham, Switzerland, 2022.
47. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Masked Feature Residual Coding for Neural Video Compression

Chajin Shin [1], Yonghwan Kim [1], KwangPyo Choi [2] and Sangyoun Lee [1,*]

[1] School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Republic of Korea; chajin@yonsei.ac.kr (C.S.); ywkimblue@yonsei.ac.kr (Y.K.)
[2] Samsung Seoul R&D Campus, Seoul 06765, Republic of Korea; kp5.choi@samsung.com
* Correspondence: syleee@yonsei.ac.kr

Abstract

In neural video compression, an approximation of the target frame is predicted, and a mask is subsequently applied to it. Then, the masked predicted frame is subtracted from the target frame and fed into the encoder along with the conditional information. However, this structure has two limitations. First, in the pixel domain, even if the mask is perfectly predicted, the residuals cannot be significantly reduced. Second, reconstructed features with abundant temporal context information cannot be used as references for compressing the next frame. To address these problems, we propose Conditional Masked Feature Residual (CMFR) Coding. We extract features from the target frame and the predicted features using neural networks. Then, we predict the mask and subtract the masked predicted features from the target features. Thereafter, the difference is fed into the encoder with the conditional information. Moreover, to more effectively remove conditional information from the target frame, we introduce a Scaled Feature Fusion (SFF) module. In addition, we introduce a Motion Refiner to enhance the quality of the decoded optical flow. Experimental results show that our model achieves an 11.76% bit saving over the model without the proposed methods, averaged over all HEVC test sequences, demonstrating the effectiveness of the proposed methods.

Keywords: neural video compression; deep learning; residual; mask; feature; conditional coding

1. Introduction

As higher-resolution videos are captured by camera sensors, the strain on device storage has increased significantly. Consequently, video compression has become important for reducing file size while preserving visual quality. To reduce this load, various standardized video compression technologies have been developed, which significantly reduce file sizes while maintaining video quality. Traditional video compression standards, such as H.264 [1], H.265 [2], VP9 [3], AV1 [4], and VVC [5], include numerous hand-crafted methods like block partitioning, inter and intra prediction, transformation, in-loop filter, and entropy coding. However, since each module is optimized individually and cannot leverage complex nonlinear operations, these methods are considered sub-optimal.

Recently, neural networks have shown significant performance improvements in many image and video tasks [6–13], and they have also been integrated into video compression. The notations and terminology related to neural video compression are listed in Table 1. DVC [14] replaces all modules in the traditional video codec with neural networks and trains the model in

an end-to-end manner. DVC uses a simple subtraction operation to obtain the residual between the target frame (x_t) and the predicted frame (x_c). This can be demonstrated in $H(x_t - x_c)$, which is referred to as Residual Coding. However, using a simple subtraction operation is not optimal. DCVC [15] uses x_c as a conditional input along with the target frame into the encoder in the feature domain ($H(x_t|x_c)$) which is called Conditional Coding. This allows the encoder to learn a more optimal operation to subtract the information in the predicted frame from the target frame. Brand et al. [16] assume that the entropy of $x_t - x_c$ is less than or equal to that of x_t, and show that Conditional Residual Coding ($H(x_t - x_c|x_c)$) has less entropy than or equal to that of Conditional Coding. They feed the difference between x_t and x_c into the encoder along with the conditional input (x_c). However, a simple subtraction can result in a large residual. The upper left graph in Figure 1 shows an example of the target and the prediction signal. Then, we can obtain a (a) "Simple Residual" signal, which shows a large residual. Recently, Chen et al. [17] proposed Masked Conditional Residual Coding ($H(x_t - m \odot x_c|x_c)$). They predicted a soft mask, applied it to the predicted frame (x_c), and then subtracted it from the target frame. This approach allows for a soft decision between using Conditional Coding and Conditional Residual Coding, which enables us to achieve smaller residuals. If we focus on the (b)" Masked Residual" signal in Figure 1, assuming the mask is ideal, we can obtain a much smaller residual than that in (a). However, Masked Conditional Residual Coding has two limitations. First, applying a mask in the pixel domain and subtracting it from the target frame cannot consider the diverse and high-level representations in the images. Second, it is effective in reducing error propagation to use the reconstructed features as references to compress the subsequent frame. This is because abundant temporal context information [18–21] can be included in the reference features. However, Residual-type Conditional Coding methods require decoding the bitstream to the pixel level residuals ($\hat{x}_r$) to obtain full information of the reconstructed target frame ($\hat{x}_t = \hat{x}_r + x_c$). Thus, they cannot utilize the reconstructed features of the previous frame as a reference.

Table 1. Summary of notations and terminology used in this paper.

Symbol	Full Name	Explanation
x_t	Target frame	The current frame to be encoded.
x_c	Compensated frame/Prediction	The frame predicted from previous frames with motion compensation.
m	Mask	A soft mask that determines how much of the predicted pixel or features to retain.
$\hat{x}_t$	Reconstructed frame	The decoded current frame.
$\hat{F}_t$	Reconstructed features	The decoded current features. This can be mapped to $\hat{x}_t$ with a neural network.
$C_{N\times}$	Conditional input	$1/N$ resolution features predicted by the reference frame or features with motion vector.
$\hat{x}_{t-1}$	Reference frame	The previously decoded frame used to encode the target frame.
$\hat{f}_{t-1}$	Reference features	The previously decoded features used to encode the target frame.

To address these problems, we propose Conditional Masked Feature Residual (CMFR) Coding. As shown in the signals on the bottom left in Figure 1, the target and predicted signals can be decomposed into low and high frequencies. Then, by predicting different masks for each frequency component and applying them to the prediction signals, a much smaller residual can be obtained, as shown in (c) "Decompose Residual". However, it is difficult to find the optimal components to extract for each content of the image, and the

corresponding masks are also complex. Therefore, instead of using hand-crafted methods to extract the components from the image, we use a neural network to extract various features from the target frame and predicted features. We also predict the masks for each channel of the features using another neural network. Using these neural networks, our CMFR Coding can be expressed as $H(F_t(x_t) - m \odot F_c(x_c)|x_c)$, where F_t and F_c represent the neural networks.

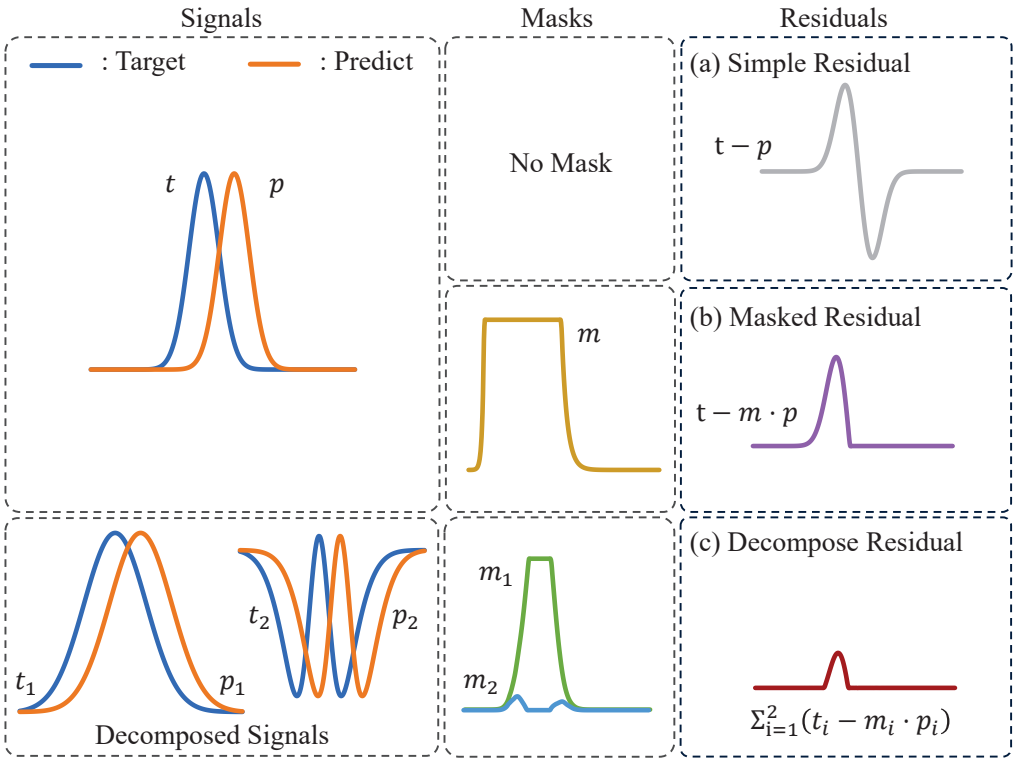


Figure 1. Upper Left: Target and Predict signals. Bottom Left: Decomposing each signal into low and high frequencies. To obtain the low-frequency signals, $t1$ and $p1$, we utilize a mean filter, while we subtract the low-frequency signals from the original signals to obtain the high-frequency signals, $t2$ and $p2$.

Moreover, we propose the Scaled Feature Fusion (SFF) module. This module predicts the fusion features and scales using conditional information along with the target frame information in the encoder or the residual information in the decoder. We apply the scales to the fusion features and subtract or add the scaled fusion features to the target or residual information, respectively. With this structure, conditional information can be effectively subtracted or added to the target or residual information within the context encoder and decoder. Furthermore, many methods of Conditional Coding [15,18–21] focus on accurately compensating the reference frame using the decompressed motion. However, they often overlook the quality of the decompressed motion. We propose a Motion Refiner module that uses reference features and the decompressed motion to improve the quality of the motion. Through various experiments, we demonstrate that the proposed methods significantly improve the rate-distortion performance of video compression.

Our main contributions can be summarized as follows:

- We point out the limitations of the existing Residual-type Conditional Coding, which masks and subtracts the prediction in the pixel domain.
- To resolve this problem, we propose a CMFR Coding, which extracts the features from the image followed by masking and subtraction in the feature domain.

- We propose the SFF module to effectively subtract or add conditional information to the target or residual information in the context encoder or decoder, respectively.
- We introduce the Motion Refiner module to enhance the quality of the decompressed motion vectors.

2. Related Works

Recently, deep learning techniques have demonstrated significant performance improvements in video processing, and there are many attempts to apply these techniques to neural video compression. Basically, they use a neural image compression structure [22–24] to compress the motion and the residual information. Lu et al. [14] propose the DVC model, which replaces all modules in the traditional codec with a neural network and trains the model in an end-to-end manner. They achieve a performance comparable to x265 [25] with the "`veryfast`" option, which is one of the fast encoding presets. They predict the motion between the reference frame ($\hat{x}_{t-1}$) and the target frame, x_t, and compress it using the neural image compression structure [24]. Then, they predict the approximation of the target frame using the motion vector. Subsequently, the difference between the target frame x_t and the prediction (x_c) is encoded using a simple subtraction operation in the pixel domain, referred to as Residual Coding ($H(x_t - x_c)$). Using this residual structure, many papers [26–29] compress the residual between x_t and x_c.

DCVC [15] highlights that a simple subtraction operation to obtain the residual is sub-optimal. To solve this problem, DCVC proposes Conditional Coding ($H(x_t|x_c)$). DCVC extracts features from the reference frame and warps them in the feature domain to obtain the predicted feature of the current frame. Then, the target frame is fed into a context encoder, using the predicted feature as conditional input. With this approach, the context encoder can learn an optimized nonlinear function that subtracts the information in the predicted features from the target frame, achieving higher rate-distortion performance than the Residual Coding methods. Moreover, various works [18–21] that use Conditional Coding predict the approximation of the target frame in the feature domain at multi-scales, and utilize them as conditional input for the context encoder. This structure shows better performance than using only the predicted features of the original resolution.

However, the information bottleneck effect [30] can occur in the Conditional Coding approach due to information loss during the extraction of the features from the reference frame (x_{t-1}) to obtain the conditional input features of the context encoder. Consequently, it may not be possible to reduce the residual entropy to zero even if the result of motion compensation is identical to the target frame. To address this issue, Brand et al. [16] propose a new scheme to obtain the residual called Conditional Residual Coding ($H(x_t - x_c|x_c)$). They assume that the entropy of $x_t - x_c$ is smaller than or equal to that of x_t, and demonstrate that Conditional Residual Coding ($H(x_t - x_c|x_c)$) has less entropy than or equal to that of Conditional Coding. For processing the Conditional Residual Coding, they obtain the difference between x_t and x_c using a simple subtraction operation. Then, the difference is fed into the context encoder along with the conditional information, x_c. However, a simple subtraction operation can lead to large residuals, especially when the prediction of high-frequency components is incorrect.

To address this issue, Chen et al. [17] propose a new scheme called Masked Conditional Residual Coding ($H(x_t - m \odot x_c|x_c)$). If the mask (m) has a value of 0, the encoder compresses the residual using Conditional Coding. In contrast, if the mask value is 1, the residual is encoded into the bitstream using Conditional Residual Coding. Therefore, they use a soft decision process with the mask to select which methods to use for encoding. With this structure, they achieve significantly better performance than other coding methods.

Meanwhile, Mentzer et al. [31] do not utilize motion information, and instead implicitly remove redundant information along the temporal axis using a transformer [32]-based entropy model. However, this process is performed only at the 1/16 scale. Lu et al. [33] use a hierarchical structure to implicitly reduce temporal redundancy. Nevertheless, the performance of their model is still lower than that of other state-of-the-art models because they also do not utilize motion vectors. Jia et al. [34] use implicit temporal modeling with simple convolutional layers and still outperform VVC by 21% in BD-rate. Jiang et al. [35] use both motion vectors and a transformer structure to capture correlations in the temporal context, particularly focusing on non-local correlations among the frames, and achieve a 38% bit saving on the UVG [36] dataset. Recently, bi-directional referencing has also been introduced in neural video compression models [37–39] to process the random access mode.

3. Method

In this section, we first describe the overall structure and then present the details of each proposed module.

3.1. Overall Structure

The goal of neural video compression is to compress the target frame by removing temporal redundancy with previous frames and encoding only the residual information. To do this, we estimate the motion vector (v_t) between the reference frame (x_{t-1}) and the target frame (x_t). We estimate v_t using pre-trained SPyNet [40], which is widely used by existing neural video compression methods [14,15,18–21]. This motion vector is compressed and reconstructed ($\tilde{v}_t$) using a motion encoder and decoder. This encoder has four convolutional layers with a kernel size of 3×3 and a stride of 2 for downsampling, and residual blocks consisting of two 3×3 convolutional layers with Leaky ReLU [41]. Moreover, the qp step (qs) is multiplied by the intermediate features at the last layer of each resolution for quality control. The qp step has the shape of $qs \in \mathbb{R}^{1\times C\times 1\times 1}$, where C represents the channel size. The motion decoder has a symmetric structure with the encoder except that it uses convolutional layers with a kernel size of 1×1 followed by a pixel shuffle [42] for upsampling. Then, we employ a Motion Refiner module to enhance the motion vector. The refined motion vector ($\hat{v}_t$) is used with the Multi-scale Context Fusion module to predict the approximation of the target frame in the feature domain at the original resolution, as well as at 1/2 and 1/4 resolutions ($C_{1\times}$, $C_{2\times}$, and $C_{4\times}$). We use a Temporal Context Mining (TCM) module from [18] as the Multi-scale Context Fusion module to predict ($C_{1\times}$, $C_{2\times}$, and $C_{4\times}$). The predicted features at the original resolution ($C_{1\times}$) are transformed and masked using a Mask Prediction module. Thereafter, the residual is obtained by subtracting these masked features from the features extracted from the target frame in the Residual Generator module. We compress this residual by feeding it into the context encoder, using the multi-scale predicted features as conditional input. In the context encoder and decoder, we introduce the SFF module to more effectively subtract information in the conditional input from the features of the target frame.

On the decoder side, we reconstruct the residual features ($\hat{r}_t$) which are combined with f_{sc} in the Combiner module to obtain the reconstructed features ($\hat{f}_t$). Finally, we transform $\hat{f}_t$ into the reconstructed target frame ($\hat{x}_t$) through a Pixel Mapping block composed of a single convolutional layer. $\hat{f}_t$ contains abundant temporal context information to prevent error propagation. Therefore, we store $\hat{f}_t$ and $\hat{x}_t$ in the reference buffer to utilize them when compressing the next frame.

We set the base channel size of the motion encoder and decoder to 64 for all resolutions. In contrast, the base channel sizes for all other modules are set to 48, 64, 96, 96, and 128 at 1/1, 1/2, 1/4, 1/8, and 1/16 resolutions, respectively. In addition, to obtain the qp step vector, we interpolate between the learnable parameters of $qs_{\max} \in \mathbb{R}^{1\times C\times 1\times 1}$ and $qs_{\min} \in \mathbb{R}^{1\times C\times 1\times 1}$ using the following equation:

$$qs = \alpha \cdot qs_{\max} + (1-\alpha) \cdot qs_{\min}, \tag{1}$$

where α is the constant obtained by log-scale interpolation in the range from 0 to 1.

3.2. Motion Refiner

After estimating the motion vector between the reference frame and the target frame, we obtain the reconstructed motion vector ($\tilde{v}_t$). Many Conditional Coding models [15,18–21] focus on warping and compensating the reference frame or features to accurately predict the target frame. They predict the features of $C_{1\times}$, $C_{2\times}$, and $C_{4\times}$, which are approximations of the target frame at the original resolution, 1/2 resolution, and 1/4 resolution, respectively. However, they often overlook the quality of the reconstructed motion vector.

The reconstructed motion vector ($\tilde{v}_t$) often produces blurred edges around moving objects. We refine this motion before performing warping and compensation at the multi-scale level. First, we warp the reference feature using the reconstructed motion vector. Then, we use the motion vector and the warped features to compute Δv_t.

$$\Delta v_t = V_{\theta v}(\tilde{v}_t, W(\hat{f}_{t-1}, \tilde{v}_t)), \tag{2}$$

where $V_{\theta v}$ denotes the neural network with learnable parameters θv, and W represents the backward warping operation. Subsequently, we obtain the refined motion vector ($\hat{v}_t$) by adding Δv_t to $\tilde{v}_t$.

$$\hat{v}_t = \tilde{v}_t + \Delta v_t \tag{3}$$

Using this refined motion vector, we warp and compensate the reference features to predict the approximate features of the target frame ($C_{1\times}$, $C_{2\times}$, and $C_{4\times}$). We use these approximate features as conditional input for both the context encoder and the decoder.

3.3. Conditional Masked Feature Residual Coding

Masked Conditional Residual Coding [17] ($H(x_t - m \odot x_c | x_c)$) is processed in the pixel domain. Therefore, it is not possible to apply different masks to the diverse and high-level representations of the image. To address this problem, we introduce CMFR Coding, which extracts features from the image, applies masking, and performs the subtraction.

First, we use a neural network to extract features from the target image ($F_t(x_t)$). Second, we predict the mask using a neural network. To predict the mask, we use the predicted features at the original resolution ($C_{1\times}$), the refined motion vector, and the warped reference frame.

$$s = S_{\theta s}(C_{1\times}, \hat{v}_t, W(\hat{x}_{t-1}, \hat{v}_t)), \tag{4}$$

where $S_{\theta s}$ denotes the neural network with learnable parameters θs. Then, we map $C_{1\times}$ to $F_{c,C_{1\times}}$ using a mapping function (F_c), and use a tanh function [43] to mask $F_{c,C_{1\times}}$ as shown in the following equation:

$$\begin{aligned} f_{sc} &= 0.5 \cdot [F_{c,C_{1\times}} + F_{c,C_{1\times}} \odot \tanh(s)] \\ &= 0.5 \cdot (1 + \tanh(s)) \odot F_{c,C_{1\times}}, \end{aligned} \tag{5}$$

where $\odot$ represents element-wise multiplication. On the context encoder side, the Residual Generator module is used to obtain the feature residual (r_t) by subtracting f_{sc} from the feature extracted from x_t.

$$r_t = F_t(x_t) - f_{sc} \tag{6}$$

Thereafter, we feed r_t into the context encoder along with the conditional inputs predicted by the Multi-scale Context Fusion module.

$$y_t = \mathrm{Enc}(r_t, C_{1\times}, C_{2\times}, C_{4\times}), \tag{7}$$

where y_t represents the latent vector. We refer to this residual encoding method as CMFR Coding.

$$H(F_t(x_t) - m \odot F_c(x_c)|x_c) \tag{8}$$

In contrast, on the Context Decoder side, we use the Combiner module to obtain the reconstructed features of the target frame. The Context Decoder predicts the reconstructed feature residual ($\hat{r}_t$). Then, we add f_{sc} to $\hat{r}_t$ and apply a residual block to obtain the reconstructed features.

$$\hat{f}_t = \mathrm{RB1}(\hat{r}_t + f_{sc}), \tag{9}$$

where RB1 denotes the residual block consisting of two convolutional layers with a kernel size of 1×1 and a Leaky ReLU [41] activation function. Subsequently, we utilize a convolutional layer to transform $\hat{f}_t$ into the pixel domain, producing $\hat{x}_t$. This approach allows us to store the reconstructed features ($\hat{f}_t$), which contain abundant temporal context information, in the reference buffer.

3.4. Scaled Feature Fusion

The context encoder transforms only residual information, which is the difference between the target frame and the conditional input, into the latent vector (y_t). To achieve this, we downsample the input features (r_t), and concatenate them with the predicted condition features ($C_{1\times}$, $C_{2\times}$, and $C_{4\times}$) at each resolution. To more effectively subtract conditional information from the input features, we introduce the SFF module.

First, we predict the fusion features (u) and scales (a) using the input features and the predicted condition features as follows:

$$\begin{aligned} u &= U_{\theta u}(r_{n\times}, C_{n\times}) \\ a &= \tanh(A_{\theta a}(r_{n\times}, C_{n\times})), \end{aligned} \tag{10}$$

where $U_{\theta u}$ and $A_{\theta a}$ represent neural networks to predict the fusion features and the scales, respectively. $r_{n\times}$ denotes the first features at $1/n$ resolution from the context encoder or decoder. These scales (a) operate through an attention mechanism, allowing it to focus on the regions that need to be subtracted effectively. Subsequently, we apply the scales to the fusion features and feed them into a convolutional layer to match the channel size of $r_{n\times}$. Then, we subtract the output of the convolutional layer from $r_{n\times}$, concatenate the result with $C_{n\times}$, and feed the combined features into a residual block.

$$\begin{aligned} o_1 &= r_{n\times} - \mathrm{Conv}(u \odot a) \\ o_2 &= \mathrm{DRB}(\mathrm{Cat}(o_1, C_{n\times}))), \end{aligned} \tag{11}$$

where Cat denotes the concatenation operation. DRB represents the depth-wise residual block, as depicted in Figure 2 (D-ResBlk). This module is used in both the context encoder and decoder at $1/2$ and $1/4$ resolutions.

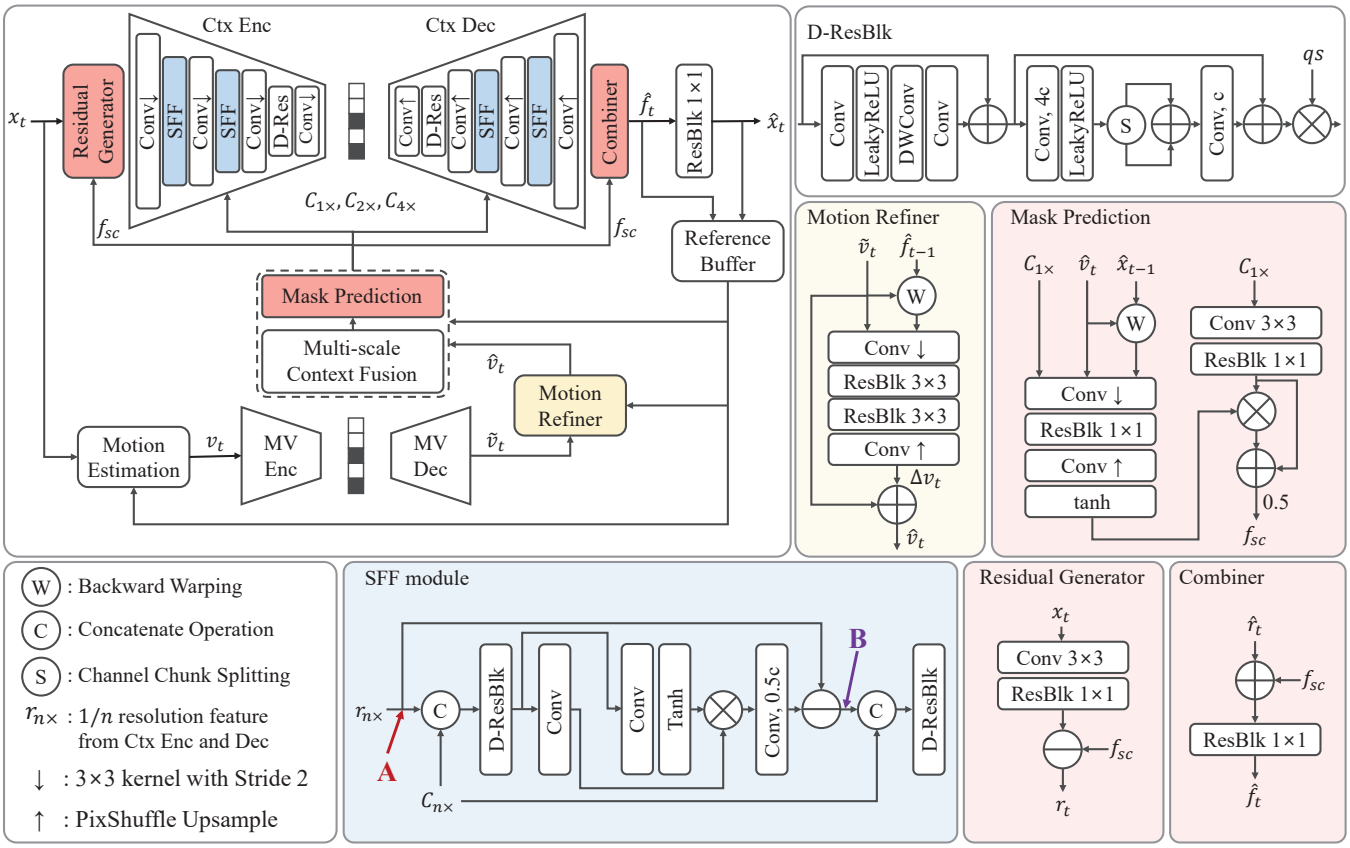


Figure 2. Overall structure of the network. The module diagrams colored in red, blue, and yellow represent the structures of our proposed Conditional Masked Feature Residual (CMFR) Coding, Scaled Feature Fusion (SFF) module, and Motion Refiner, respectively. "Conv" without any additional mark refers to a 1×1 convolutional layer, while "Conv, nc" represents the 1×1 convolutional layer with output channel size multiplied by a factor of n. "DWConv" denotes the depth-wise convolutional layer with a kernel size of 3×3. "ResBlk $k \times k$" denotes the residual block consisting of two $k \times k$ convolutional layers with a LeakyReLU [41].

4. Experiments

4.1. Implementation Details

To train our model, we use the Vimeo-90K [44] dataset, which consists of approximately 90,000 sequences, each containing seven frames. Each frame has a resolution of 448×256. During the training stages, we randomly crop frames to a size of 256×256 and apply random vertical and horizontal flips. Additionally, we randomly reverse the order of the frames in each sequence. We train our model with the multi-stage training strategy used in DCVC-TCM [18]. Unlike DCVC-TCM, we modify the training process in terms of compression quality control. We find that training our model becomes unstable when variable rate control modules are included from the beginning. Therefore, we first train our model at the highest video quality. After training at the highest quality, we incorporate variable rate control modules to enable training across varying compression levels. Then, to mitigate error propagation, we use a cascaded loss [18] to train the entire model on consecutive multi-frames.

In the training stage with multi-frames, we use six P-frames instead of the five that DCVC-TCM used. For the loss function, we use the Lagrange multiplier λ, which balances the quality of the reconstructed frame and the bit rate in a random range of $[320, 2600]$. In addition, to reduce error propagation and maintain compression performance over

long temporal sequences, we fine-tune our model using the BVI-DVC [45] dataset. This dataset consists of approximately 800 sequences at various resolutions. We fine-tune our model using 36 consecutive frames with cascaded loss [18] for 4 epochs at a learning rate of 1×10^{-5}. Then, we decrease the learning rate to 1×10^{-6} for 1 epoch. For the MS-SSIM distortion, we use the $1 - \text{MS-SSIM}(\hat{x}_t, x_t)$ distortion function to train our model with λ values from 10 to 81.

4.2. Evaluation

To evaluate the video compression performance of each model, we use the UVG [36] dataset, which consists of seven sequences at a resolution of 1920×1080. Moreover, we use HEVC Class A to E [46], which have resolutions of 2560×1600, 1920×1080, 832×480, 416×240, and 1280×720, respectively. Additionally, we utilize the MCL-JCV [47] dataset. This dataset consists of 30 sequences with a resolution of 1920×1080. For all test datasets, we apply reflection padding to each input sequence to ensure that the width and height of the frames are multiples of 64. We use an intra period of 32 and a total of 96 frames for compression during the test.

To compare the rate-distortion performance of our model, we use HEVC [2], VP9 [3], and VVC [5] for traditional codec methods. In the case of HEVC, we use the HEVC reference software (HM) [48] version 18.0. For VP9, we use FFmpeg [49] version 7.0 with libvpx-vp9 of version 1.14.0. In the case of VVC, we use the VVC test model (VTM) [50] of version 23.4. For neural network-based P-frame video compression methods, we compare our model with DCVC [15], DCVC-TCM [18], DCVC-HEM [19], and DCVC-DC [20]. To evaluate rate-distortion performance, we measure PSNR and bits per pixel (BPP) and plot these values on a graph. Therefore, the graph on the left is a better model, as it uses fewer bits for the same visual quality. Furthermore, we utilize the BD-rate [51] to confirm the average bit savings achieved at the same visual quality level.

4.3. Rate-Distortion Performance

Figures 3 and 4 show the rate-distortion curves. The x-axis and y-axis represent the bpp and PSNR, respectively. Our model outperforms all traditional video codec methods, DCVC, DCVC-TCM, and DCVC-HEM, across all test sequences. Moreover, our model demonstrates higher compression performance in most sequences compared to DCVC-DC. Table 2 shows the BD-rate using VTM as an anchor with the PSNR distortion function, the total number of parameters, and the encoding and decoding time. To measure inference time, we use sequences with a resolution of 1920×1080 and average the results to obtain milliseconds per frame. For GPU measurements, we use the RTX A6000 Ada, while we utilize Ryzen 5 7500F for the CPU. The results indicate that our model demonstrates better compression performance than other models across most of the test sequences. Furthermore, compared to the DCVC-DC model, our model shows higher performance while requiring fewer parameters and less time for decoding. Table 3 represents the BD-rate with the MS-SSIM distortion function. As shown in the table, our model also achieves higher performance than others on most test sequences under the MS-SSIM metric.

Figure 5 presents the qualitative comparison results. As shown in the upper image, our model preserves the pattern of the floor tiles better (blue box) and retains the folding lines of the parasol more clearly (red box). In the lower image, the texture of the girl's hair is better preserved in our result (blue box). Moreover, the dot below the number 4 on the die is clearer than in the results of the others. In addition, the structure of the wrinkles on the orange bag is better maintained (green box).

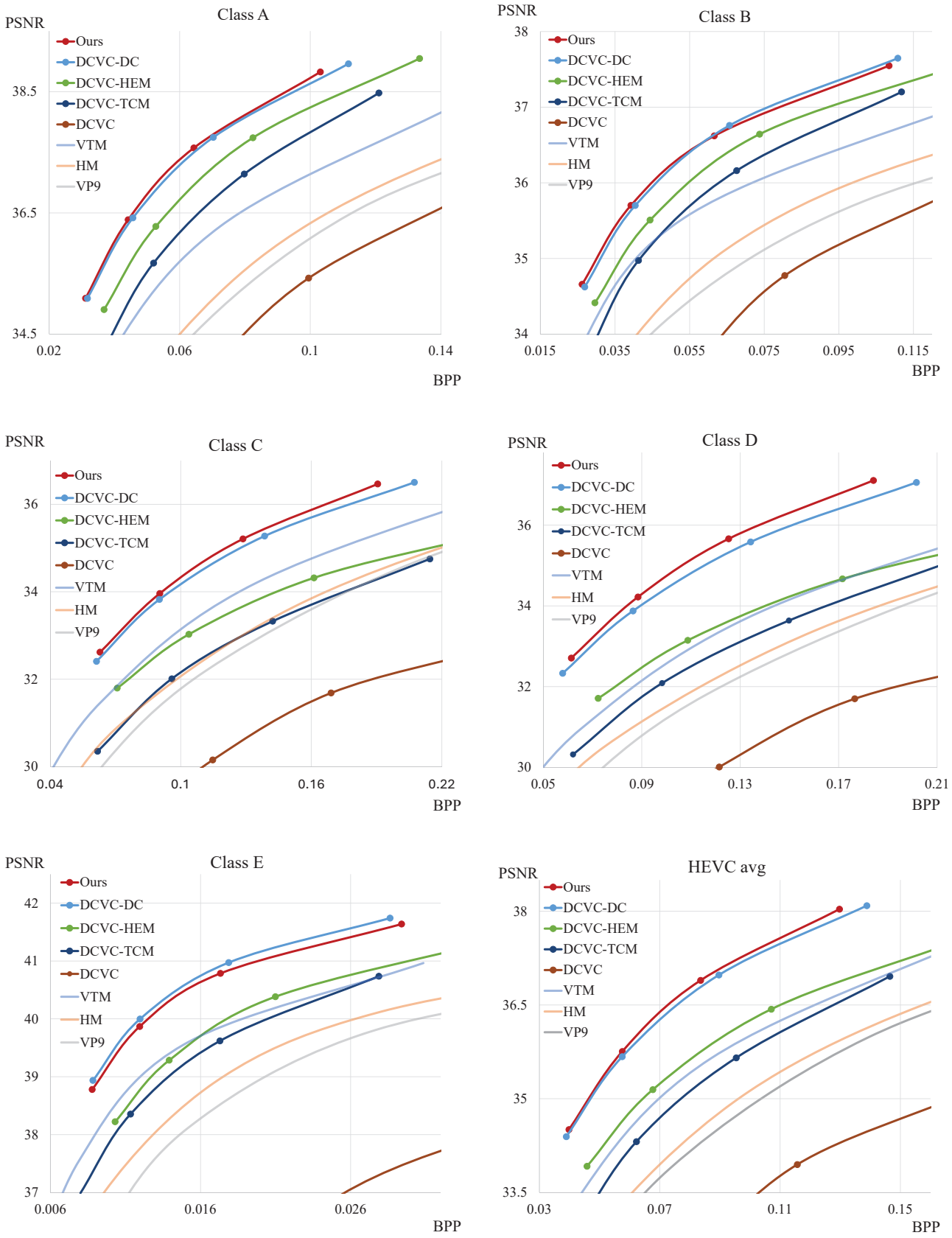


Figure 3. Rate-distortion performance comparison for different test sequences against several models.

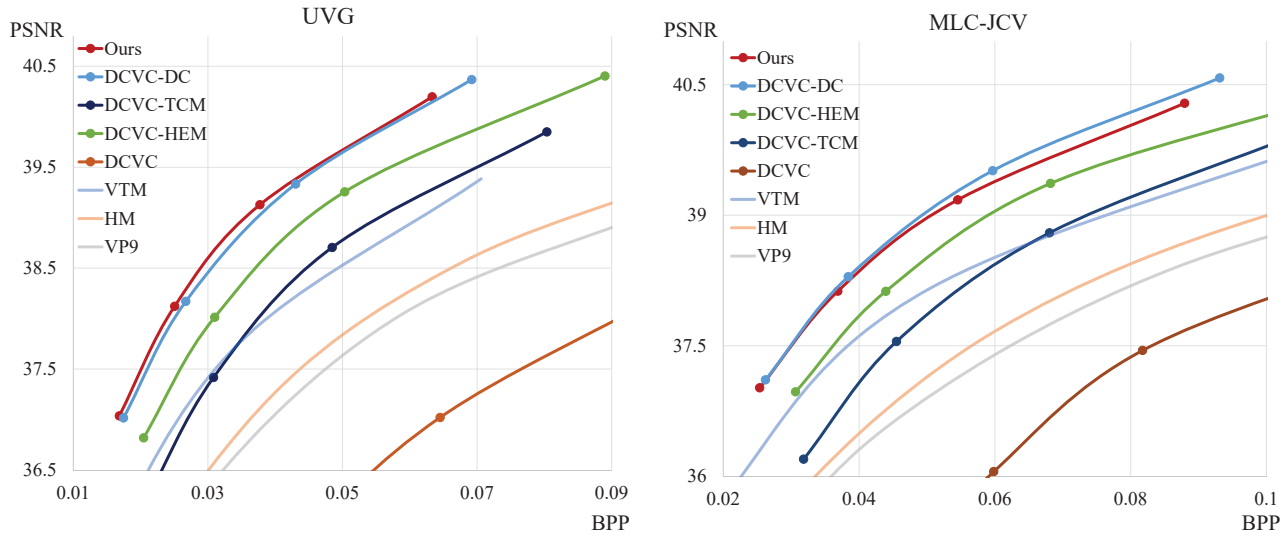


Figure 4. Rate-distortion performance comparison for different test sequences against several models.

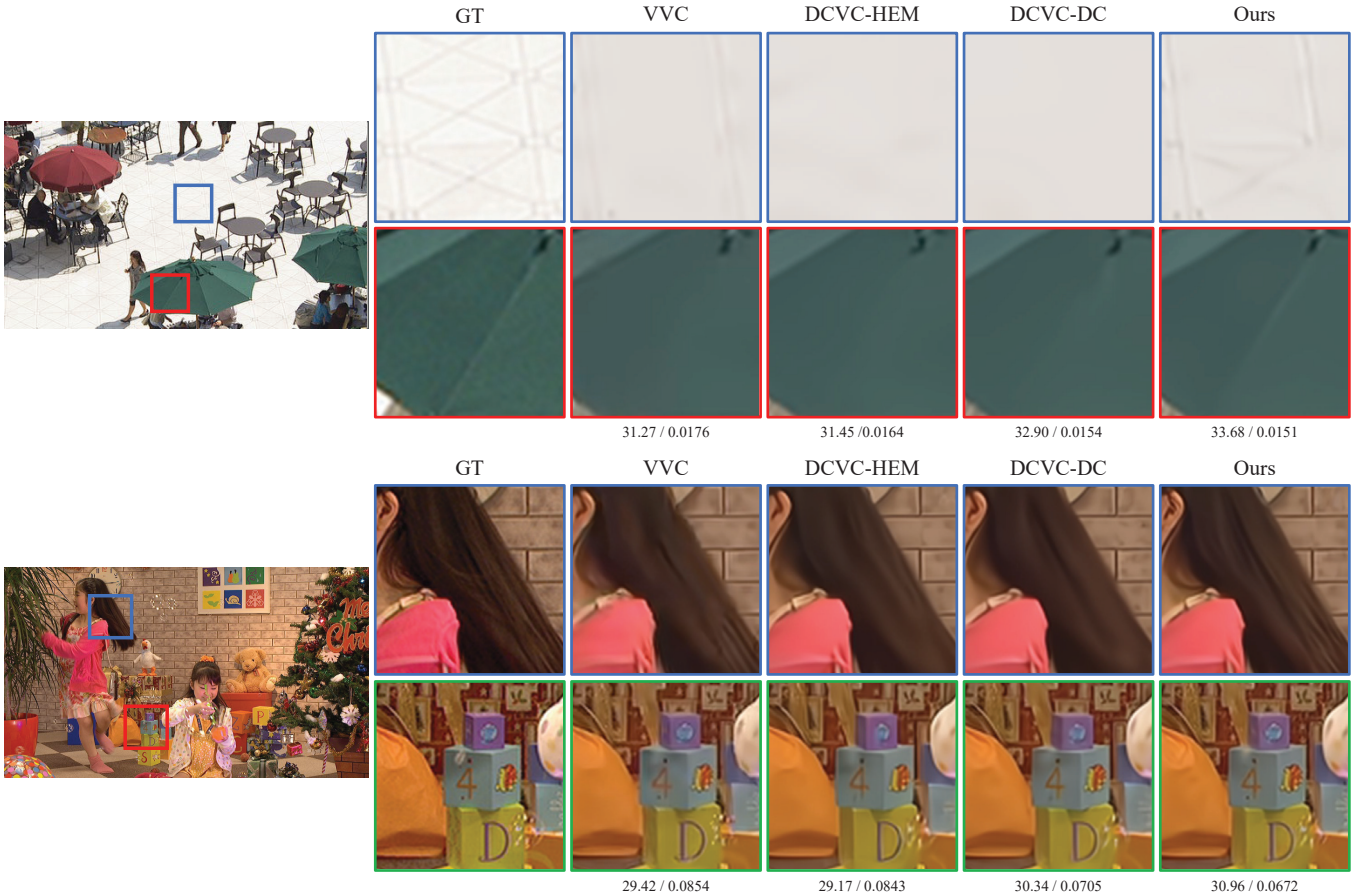


Figure 5. Qualitative comparison results with other SoTA neural video compression models and traditional codec. The values under the images indicate PSNR and BPP, respectively.

Table 2. Rate-distortion performance comparison with other models across various test sequences on PSNR distortion. Inference time is measured using sequences with a resolution of 1920 × 1080.

Model	Dataset	VP9	HM	VTM	DCVC	DCVC-TCM	DCVC-HEM	DCVC-DC	Ours
BD-rate (%)	UVG	54.81	40.82	0.00	168.37	0.27	−19.74	−35.06	−37.90
	HEVC-A	48.77	36.67	0.00	85.81	−14.66	−27.74	−39.67	−41.09
	HEVC-B	65.06	44.72	0.00	131.66	−3.56	−18.49	−32.07	−33.13
	HEVC-C	40.99	30.76	0.00	162.16	33.04	11.53	−24.07	−27.48
	HEVC-D	39.08	29.83	0.00	136.30	12.97	−1.97	−37.01	−41.37
	HEVC-E	67.05	45.01	0.00	285.24	13.43	1.15	−33.68	−30.08
	HEVC-avg	46.69	35.65	0.00	146.61	10.21	−6.38	−34.28	−36.28
	MCL-JCV	57.74	42.94	0.00	139.63	10.40	−13.82	−29.35	−27.34
Parameters (M)		-	-	-	7.94	10.71	17.52	19.78	19.48
GPU Enc Time (ms/frame)		-	-	-	2438.34	374.15	318.33	464.75	482.93
GPU Dec Time (ms/frame)		-	-	-	12,345.73	230.37	192.01	359.74	334.15
CPU Enc Time (ms/frame)		307.89	8357.62	37,905.31	17,831.36	26,427.89	23,015.48	35,378.43	37,520.83
CPU Dec Time (ms/frame)		6.41	30.71	45.8	16,446.42	23,247.03	19,253.71	27,824.46	26,073.00
KMACs/pixel (Enc)		-	-	-	1180.38	1750.62	1700.87	1360.05	1490.21
KMACs/pixel (Dec)		-	-	-	776.26	933.10	1270.31	930.52	918.71

Table 3. Rate-distortion performance comparison with other models across various test sequences on MS-SSIM distortion.

Model	Dataset	VP9	HM	VTM	DCVC	DCVC-TCM	DCVC-HEM	DCVC-DC	Ours
BD-rate (%)	UVG	32.23	18.68	0.00	57.97	−12.22	−36.08	−40.44	−42.07
	HEVC-A	35.22	24.73	0.00	71.61	−0.15	−31.46	−38.18	−37.25
	HEVC-B	33.24	23.19	0.00	55.32	−15.12	−45.74	−51.15	−52.46
	HEVC-C	28.35	22.07	0.00	40.13	−17.88	−41.57	−53.48	−55.61
	HEVC-D	30.1	25.16	0.00	18.73	−31.02	−54.20	−61.91	−63.79
	HEVC-E	32.08	28.49	0.00	104.56	−28.67	−54.57	−59.83	−57.32
	HEVC-avg	32.84	23.41	0.00	62.48	−25.4	−41.29	−53.41	−56.41
	MCL-JCV	31.57	27.63	0.00	29.06	−11.77	−36.61	−48.75	−46.03

5. Discussion

5.1. Conditional Masked Feature Residual Coding

We compare our proposed method, CMFR Coding, with other Conditional Coding approaches. Figure 6 shows the rate-distortion curves of various coding methods. "Base_NoRefFeat" denotes the Conditional Coding model that uses reference frames instead of reference features. Therefore, this model fails to utilize abundant temporal context information. By applying Conditional Residual Coding or Masked Conditional Residual Coding to this base model, we find that the performance improves by 0.92% and 1.99%, respectively. When we apply our CMFR Coding to the base model, we observe a significant performance improvement of 3.46%. "Base" represents the base model, which uses reference features to leverage abundant temporal context information. Conditional Residual Coding and Masked Conditional Residual Coding cannot utilize this base model structure. In contrast, applying our method to both the "Base" and "Base_NoRefFeat" models demonstrates that CMFR Coding consistently improves rate-distortion performance. Table 4 and Figure 7 illustrate the performance improvements achieved by applying our proposed modules, along with the increases in model parameters and encoding/decoding times. Compared to base model A, model B from Table 4 achieves a performance improvement of

3.42% with a relatively small increase in the number of parameters. However, since CMFR Coding is applied at the original resolution, the inference time increases slightly.

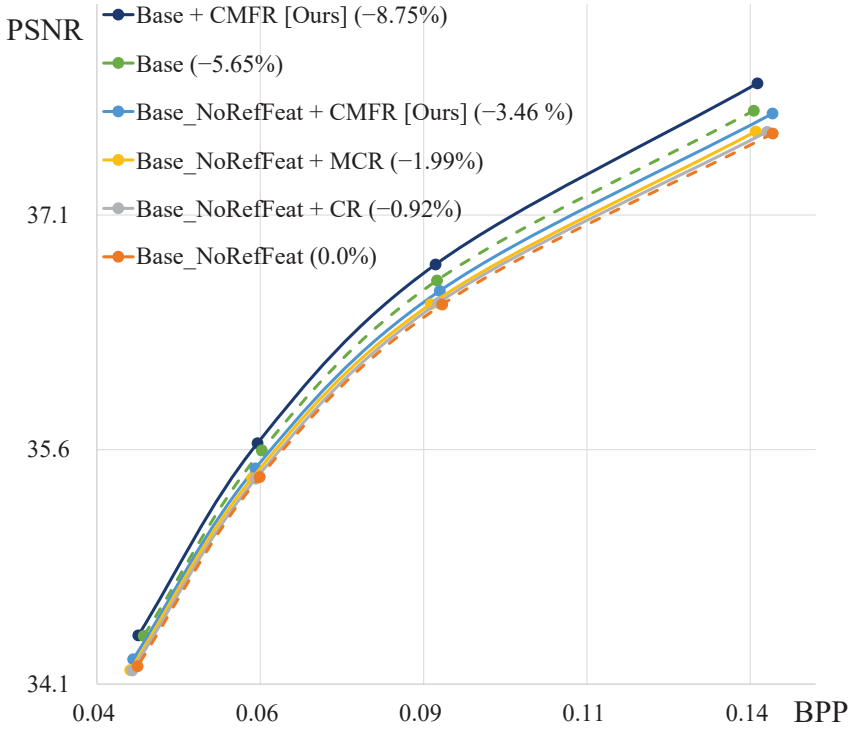


Figure 6. Ablation study results of various types of Conditional Coding on all HEVC test sequences. "NoRefFeat" indicates that we use the reference frame instead of features. "CMFR" denotes the proposed Conditional Masked Feature Residual Coding. "MCR" and "CR" represent Masked Conditional Residual Coding and Conditional Residual Coding, respectively.

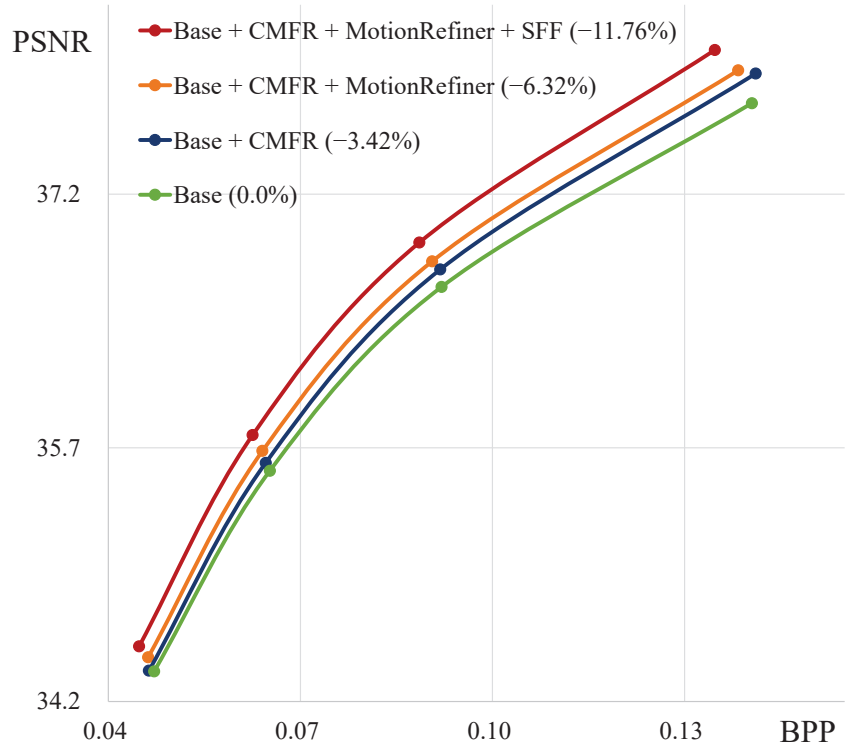


Figure 7. Ablation study result of each module on all HEVC test sequences. CMFR denotes the Conditional Masked Feature Residual Coding, and SFF represents the Scaled Feature Fusion module.

Table 4. Ablation study results of each proposed module on all HEVC test sequences. Inference time is measured using sequences with a resolution of 1920 × 1080 on the RTX A6000 ada.

	A	**B**	**C**	**D**
CMFR		✓	✓	✓
MotionRefiner			✓	✓
SFF				✓
BD-rate (%)	0.0	−3.42	−6.32	−11.76
Parameters (M)	17.41	17.64	18.34	19.48
GPU Enc Time (ms/frame)	332.79	398.66	411.18	482.93
GPU Dec Time (ms/frame)	212.48	275.73	298.03	334.15

Figure 8 shows that the masks of our CMFR Coding are applied differently for each channel. (a) in Figure 8 shows the target frame, while (b) depicts the predicted features, which are $F_{c,C_{1\times}}$ in Equation (5). (c) illustrates the masks, defined as $0.5 \cdot (1 + \tanh(s))$ in Equation (5). Finally, (d) presents the masked predicted features, denoted as f_{sc} in Equation (5). These results indicate that the mask values change significantly at the boundaries of the moving objects. Moreover, although the spatial location is the same, the masks are applied differently across channels. For the same spatial location, some channels activate with high mask values, while others deactivate with low mask values.

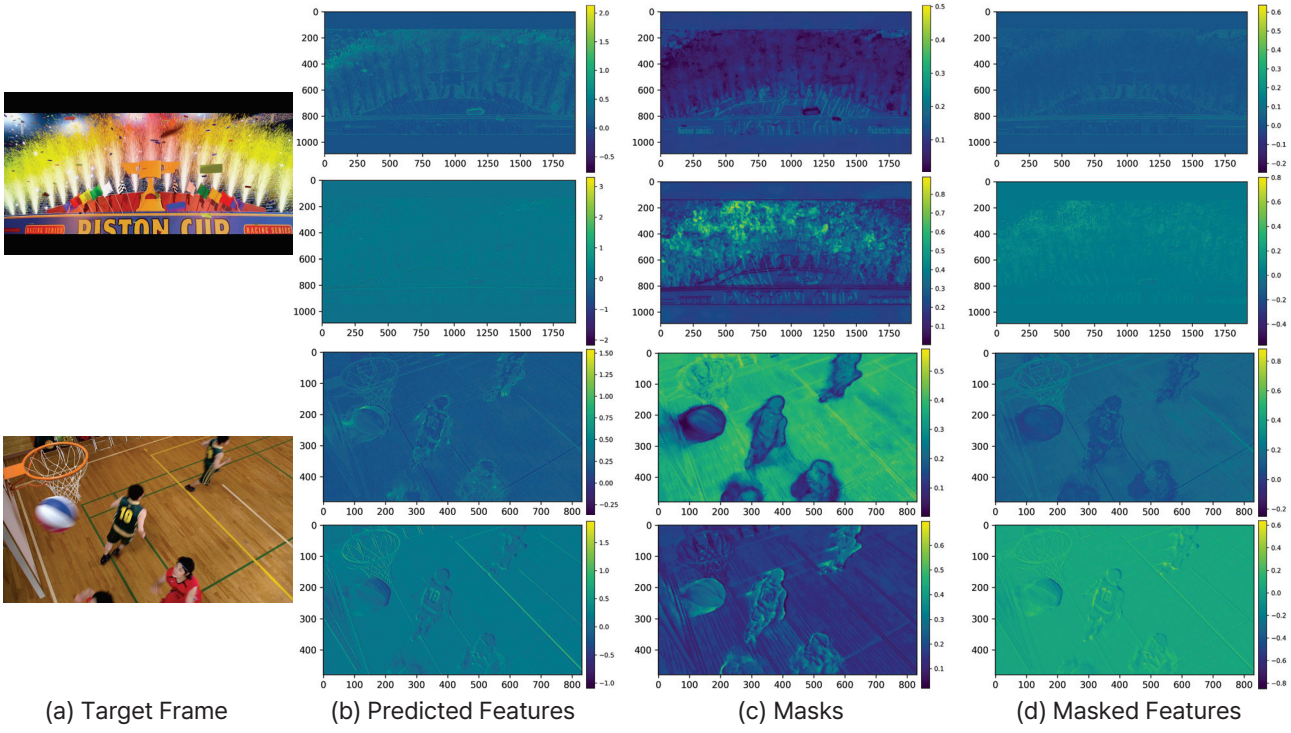


Figure 8. Visualization results of the masks and the masked predicted features from the Masked Prediction module from Figure 2 and Equation (5).

5.2. Motion Refiner

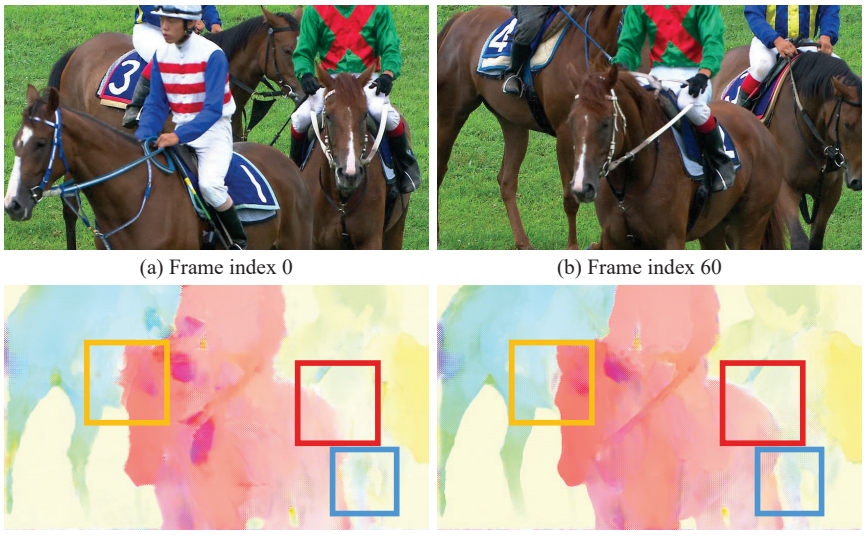
Figure 9 visualizes the results of the Motion Refiner module. This module refines the decoded motion vector using the reference features. (a) shows the decoded motion vector ($\tilde{v}_t$) obtained through the decoder, and (b) depicts the refined motion vector ($\hat{v}_t$). In (a), the edges of the moving object appear blurred. In contrast, the edges of the refined motion vector are clear and sharp. The rate-distortion plot in Figure 7 shows that the Motion Refiner module consistently improves video compression performance across the

entire quality range. Furthermore, as shown in Table 4, this module achieves a performance improvement of 2.9%, while it requires only an additional 0.7 M parameters and increases decoding time by approximately 23 ms.

(a) Decoded Motion Vector (b) Refined Motion Vector

Figure 9. (**a**) Reconstructed motion vector, which is the output of the motion decoder. (**b**) Refined motion vector.

Figure 10 demonstrates how the neural network refines the motion vector using reference features. (a) and (b) represent the frames with index 0 and 60, respectively. Moreover, (c) and (d) depict the refined motion vectors for frame index 61. (c) uses the reference feature from (a), while (d) utilizes the reference feature from (b). Even if the structure of the target motion is not aligned with the reference frame, the edge shapes remain considerably similar. The neural network can use this characteristic to refine the edge or overall shape of the motion vector. As shown in Figure 10, the motion vector of frame index 61 has a similar edge shape with the previous frame (b). Therefore, (d) demonstrates a sharper edge and more accurate structure.



(a) Frame index 0 (b) Frame index 60

(c) Refined MV index 61 with (a) (d) Refined MV index 61 with (b)

Figure 10. Motion refining analysis. (**a**) and (**b**) represent the frames with index 0 and 60, respectively. (**c**,**d**) depict the refined motion vectors for frame index 61. (**c**) uses the reference feature from (**a**), while (**d**) utilizes the reference feature from (**b**).

Figure 11 shows the results of predicting Δv_t with reference features ($\hat{f}_{t-1}$) or reference frames ($\hat{x}_{t-1}$) in the Motion Refiner module. Using $\hat{f}_{t-1}$ to predict Δv_t results in significantly better performance compared to $\hat{x}_{t-1}$. Therefore, we use reference features for motion refinement, as they are more effective.

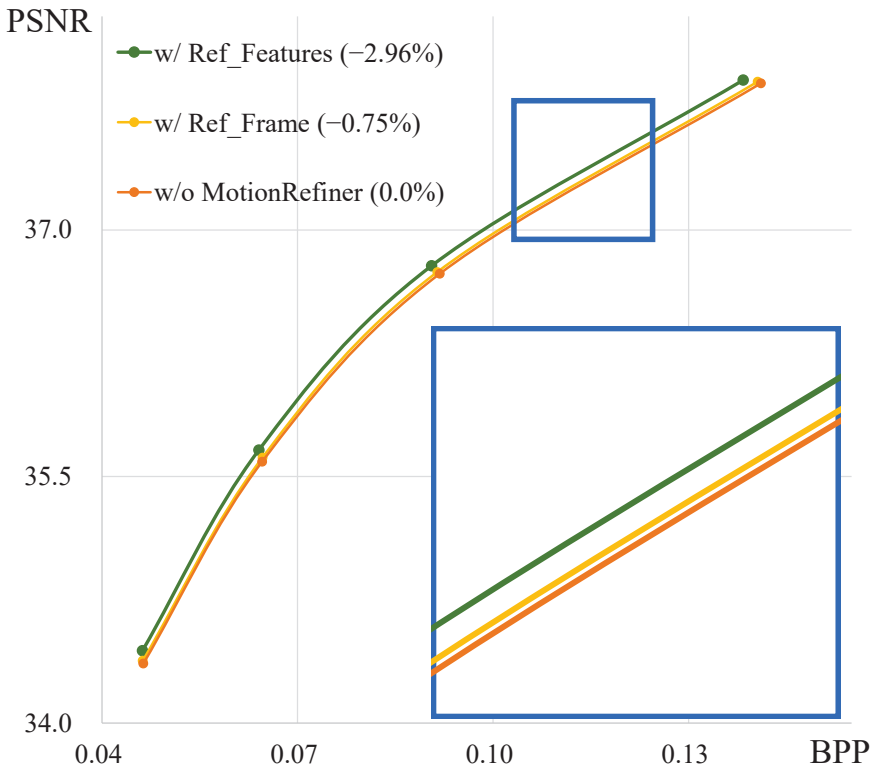


Figure 11. Rate-distortion results for motion refinement with reference features (w/ Ref_Features) or reference frame (w/ Ref_Frame) on all HEVC test sequences.

5.3. Scaled Feature Fusion

As shown in Figure 7 and Table 4, the SFF module increases performance by 5.44%, with a parameter increase of about 1.14 M. Furthermore, Table 5 demonstrates that the SFF module effectively improves rate-distortion performance. This table shows the performance with the SFF module ("w/SFF") and when it is replaced by three consecutive depth-wise residual blocks ("w/3 D-ResBlks"). These results indicate that the three consecutive residual blocks, which use more parameters than the SFF module, result in 1.46% performance increase. In contrast, our SFF module demonstrates 5.87% performance improvement while using fewer parameters.

Table 5. Ablation study result of Scaled Feature Fusion module on all HEVC test sequences. "w/SFF" represents the model with the SFF module, while "w/o SFF" denotes the model without it. In addition, "w/3 D-ResBlks" indicates the model where the SFF module is replaced with three consecutive depth-wise residual blocks.

Model	w/o SFF	w/SFF	w/3 D-ResBlks
BD-rate (%)	0	−5.87	−1.46
Parameters (M)	18.34	19.48	20.08

We visualize the intermediate features from the SFF module. To do this, we select the first SFF module in the context encoder and define two points for visualization: point A

(red) and point B (purple), as shown in the SFF module block diagram in Figure 2. Then, we freeze the pre-trained neural network and train only an additional decoder that transforms the features at points A and B into the pixel domain. This decoder consists of three and four residual blocks for points A and B, respectively, followed by a convolutional layer and a pixel shuffle layer. Using this additional decoder, we can visualize what information has been removed by transforming the intermediate features into the pixel domain. Through this process, we confirm whether the conditional information is actually subtracted from the target feature $r_{n\times}$ through the feature subtraction operation. Figure 12a shows a PSNR of 22.32 dB at point A of the SFF module in the pixel domain. After the subtraction operation with $r_{n\times}$, the PSNR is reduced to 18.36 dB at point B, as shown in (b). Moreover, as illustrated in Figure 12, the colors are distorted and many details are lost. These results confirm that the SFF module subtracts conditional information from the target feature $r_{n\times}$ through the subtraction operation.

(a) A: 22.32 dB (b) B: 18.36 dB

Figure 12. Visualization of features from the SFF module. We transform the features at point A (red) and B (purple) from the SFF module block diagram in Figure 2 into the pixel domain using additional decoder. (**a**) shows the feature at point A, and (**b**) shows the feature at point B, both transformed into the pixel domain.

5.4. Error Propagation

In this section, we evaluate the effect of error propagation and verify the effectiveness of using reconstructed features as references and fine-tuning the model with 36 consecutive frames. In the case of Residual Coding in the pixel domain, the decoder reconstructs the residual ($\hat{r}$) and then adds it to the predicted frame (x_c) to obtain the reconstructed frame ($\hat{x}_t = \hat{r} + x_c$). However, $\hat{x}_t$, which has only three channels, loses abundant temporal context information. To cope with this problem, DCVC-TCM [18] stores reconstructed features in the buffer, which contain richer temporal information across multiple channels.

To verify this, we adapt the evaluation method for error propagation utilized in DCVC-FM [21]. We use "KristenAndSara" from the HEVC test sequence class E [46] as the test sequence. Only the first frame is set as an intra frame, and the remaining 499 frames are set as P frames to induce error propagation. We evaluate the "Base" and "Base_NoRefFeat" models from Figure 6, which use reconstructed features and frame as reference, respectively. Figure 13a represents the effect of the reference type on the intensity of error propagation. The x- and y-axes represent the frame index and the PSNR value, respectively. Moreover, "Ref_Features" and "Ref_Frame" correspond to the "Base" and "Base_NoRefFeat" models, respectively. The graphs labeled "moving_avg" indicate that a moving average with a window size of 16 is applied to the PSNR values along the frame index axis. As shown in Figure 13a, "Ref_Features" shows a decrease in PSNR until around frame 100, and then it remains stable in visual quality. In contrast, the PSNR of "Ref_Frame" continues to decrease, indicating more severe error propagation.

Moreover, the Vimeo-90K [44] dataset consists of seven frames per sequence, allowing a maximum of six consecutive P-frames to be used for training, as the first frame is used as the I-frame. This leads to a significant gap between training and test settings. Therefore, it is necessary to train the network under similar conditions to the test environment in order to effectively mitigate error propagation in long-term sequences. In Figure 13b, "w/o Finetune" refers to the model trained only on the Vimeo-90k dataset, while "w/Finetune" indicates the result of further fine-tuning this model using 36 consecutive frames from BVI-DVC [45]. As shown in this figure, "w/Finetune" shows stable PSNR from around frame 100, while "w/o Finetune" shows a continuous decrease in PSNR. This observation demonstrates that reducing the gap between the training and test environments effectively mitigates error propagation.

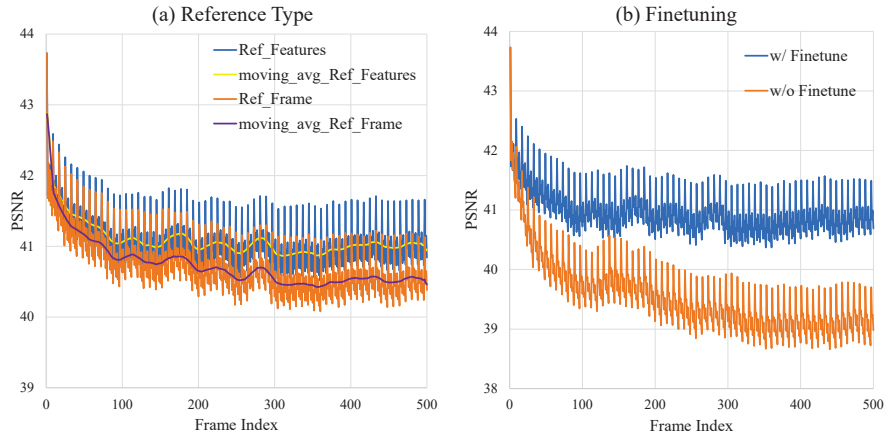


Figure 13. Visualization of error propagation effects. (**a**): Comparison between using reconstructed frames or features as reference. (**b**): Evaluating the effectiveness of fine-tuning on 36 consecutive frames using the BVI-DVC [45] dataset.

5.5. Limitations and Future Works

Although our proposed methods show performance improvements in terms of rate-distortion, several limitations remain. The first limitation is the memory size of the Decoded Picture Buffer (DPB). In the case of Conditional Residual and Masked Conditional Residual Coding, only $4 \times H \times W \times 3$ bytes are required for the size of the DPB when using 32-bit floating point. In contrast, feature-based method, including our CMFR Coding, store both the reconstructed frame and the feature maps. Therefore, when the channel size is 48, the required memory size is $H \times W \times (3 + 48) \times 4$ bytes, which is approximately 17 times larger than that of frame buffering methods. In terms of latency, there is no significant difference between storing the reconstructed frame and feature maps in terms of neural network structure. This is because both approaches transform the reconstructed frame or feature map into another feature domain using a convolutional layer and use it for motion compensation. In the case of data transfer over the bus, we measure the time required to transfer frame and feature maps from the CPU to GPU over 100 iterations. The total transfer time for frames is 0.1163 s, while the feature maps required 1.3621 s. This indicates that feature map transfer is approximately 11 times slower. However, if the reconstructed frame or feature is stored in the same device buffer as the neural network, the reference data fetching speed becomes much faster than the values we measured.

The second limitation is the large memory requirement in inference. We analyze how much GPU memory is required for decoding at commonly used resolutions. A resolution of 1280×720, which contains 921 K pixels, requires approximately 8.35 GB of video

memory. For 1920 × 1080 (2073 K pixels), the memory usage increases to about 18.24 GB, and for 2560 × 1600 (4096 K pixels), it requires 34.81 GB. In the case of 4 K resolution, which is now widely adopted, approximately 70 GB of GPU memory is required, presenting a challenge for deploying neural video compression models to edge devices. To address this issue, one of the most straightforward approaches is to reduce the bit width of the data. Replacing 32-bit floating-point precision with 16-bit or 8-bit, along with fine-tuning [52], can significantly reduce both memory usage and inference time. Additionally, techniques such as pruning [53] could be utilized in future work to further reduce memory and computational costs.

The third limitation is the long decoding time. Table 2 presents a comparison of encoding and decoding time as well as KMACs between the proposed model, other SoTA neural video compression models, and traditional codecs. For traditional codecs, unlike the neural network models, KMACs vary depending on the content, so we measure only CPU-based inference time. As shown in the Table 2, our model achieves faster encoding speed than VVC when using either GPU or CPU. However, the decoding speed is approximately 10 times slower than VVC, even with GPU acceleration. Moreover, we observe that although DCVC-DC and our model show lower KMACs than DCVC-HEM and DCVC-TCM, their inference times are longer. This is due to the network architecture, which utilize numerous 1 × 1 convolutional layers and depth-wise convolutional layers, resulting in deeper networks. These operations have relatively low computational complexity individually. However, when they are stacked and proceed sequentially, the total inference time can significantly increase, as such a sequential process cannot benefit from parallelism or fast algorithms. As a result, the inference time becomes longer despite the lower KMACs.

This long inference time leads to a significant challenge for using neural video compression models in practical environments, despite their superior performance. Therefore, reducing decoding time is critical for future works. One of the main bottlenecks in decoding is arithmetic coding performed on the CPU and data transfer between the CPU and GPU. They account for approximately 20% (65.34 ms per frame) of the total decoding time. To address this, future work could focus on performing arithmetic coding directly on the GPU or designing new architectures that allow arithmetic coding and neural network computation to be processed in parallel.

6. Conclusions

In this study, we propose Conditional Masked Feature Residual (CMFR) Coding. With this approach, we can minimize the residual information between the target frame and the compensated features. For CMFR Coding, we utilize neural networks to extract the diverse and high-level representations from the target frame. Then, we predict masks and apply them to the predicted features to reduce the residual. Moreover, we introduce the Motion Refiner, which enhances the quality of the decoded optical flow. Additionally, we propose the Scaled Feature Fusion (SFF) module to more effectively remove and add conditional information in the context encoder and decoder, respectively. We demonstrate that our methods effectively minimize residual information and achieve a 36.28% performance improvement compared to VVC on the HEVC test sequences. Furthermore, we analyze limitations of the proposed model in terms of inference time, memory usage, and buffer size.

Author Contributions: Conceptualization, C.S. and Y.K.; Methodology, C.S., Y.K. and K.C.; Software, C.S.; Validation, C.S., Y.K. and K.C.; Formal analysis, K.C.; Investigation, Y.K.; Resources, C.S.; Data curation, C.S. and S.L.; Writing—original draft preparation, C.S.; Writing—review and editing, C.S.,

Y.K., K.C. and S.L.; Supervision, S.L.; Project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Korea Planning & Evaluation Institute of Industrial Technology(KEIT) grant funded by the Korea government (MOTIE) (No. RS-2024-00442120, Development of AI technology capable of robustly recognizing abnormal and dangerous situations and behaviors during night and bad weather conditions). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. RS-2024-00423362).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicity available datasets were analyzed in this study [36,44–47]

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [CrossRef]
2. Sullivan, G.J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
3. Mukherjee, D.; Bankoski, J.; Grange, A.; Han, J.; Koleszar, J.; Wilkins, P.; Xu, Y.; Bultje, R. The latest open-source video codec vp9—An overview and preliminary results. In Proceedings of the 2013 Picture Coding Symposium (PCS), San Jose, CA, USA, 8–11 December 2013; pp. 390–393.
4. Han, J.; Li, B.; Mukherjee, D.; Chiang, C.H.; Grange, A.; Chen, C.; Su, H.; Parker, S.; Deng, S.; Joshi, U.; et al. A Technical Overview of AV1. *Proc. IEEE* **2021**, *109*, 1435–1462. [CrossRef]
5. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [CrossRef]
6. Lim, D.; Choi, J. FDI-VSR: Video Super-Resolution Through Frequency-Domain Integration and Dynamic Offset Estimation. *Sensors* **2025**, *25*, 2402. [CrossRef] [PubMed]
7. Choi, J.; Oh, T.-H. Joint Video Super-Resolution and Frame Interpolation via Permutation Invariance. *Sensors* **2023**, *23*, 2529. [CrossRef] [PubMed]
8. Kumari, P.; Keck, S.; Sohn, E.; Kern, J.; Raedle, M. Advanced Imaging Integration: Multi-Modal Raman Light Sheet Microscopy Combined with Zero-Shot Learning for Denoising and Super-Resolution. *Sensors* **2024**, *24*, 7083. [CrossRef] [PubMed]
9. Sun, J.; Yuan, Q.; Shen, H.; Li, J.; Zhang, L. A Single-Frame and Multi-Frame Cascaded Image Super-Resolution Method. *Sensors* **2024**, *24*, 5566. [CrossRef] [PubMed]
10. Lu, M.; Zhang, P. Real-World Video Super-Resolution with a Degradation-Adaptive Model. *Sensors* **2024**, *24*, 2211. [CrossRef] [PubMed]
11. Tsai, F.J.; Peng, Y.T.; Tsai, C.C.; Lin, Y.Y.; Lin, C.W. Banet: A blur-aware attention network for dynamic scene deblurring. *IEEE Trans. Image Process.* **2022**, *31*, 6789–6799. [CrossRef] [PubMed]
12. Wei, P.; Xie, Z.; Li, G.; Lin, L. Taylor neural network for real-world image super-resolution. *IEEE Trans. Image Process.* **2023**, *32*, 1942–1951. [CrossRef] [PubMed]
13. Feng, G.; Han, L.; Jia, Y.; Huang, P. Single Image Defocus Deblurring Based on Structural Information Enhancement. *IEEE Access* **2024**, *12*, 153471–153480. [CrossRef]
14. Lu, G.; Ouyang, W.; Xu, D.; Zhang, X.; Cai, C.; Gao, Z. DVC: An end-to-end deep video compression framework. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10998–11007.
15. Li, J.; Li, B.; Lu, Y. Deep contextual video compression. In Proceedings of the 34th Conference on Neural Information Processing Systems (NIPS), New Orleans, LA, USA, 10–16 December 2021; pp. 1–12.
16. Brand, F.; Seiler, J.; Kaup, A. Conditional Residual Coding: A Remedy for Bottleneck Problems in Conditional Inter Frame Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 6445–6459. [CrossRef]
17. Chen, Y.-H.; Xie, H.-S.; Chen, C.-W.; Gao, Z.-L.; Benjak, M.; Peng, W.-H.; Ostermann, J. MaskCRT: Masked Conditional Residual Transformer for Learned Video Compression. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 11980–11992. [CrossRef]
18. Sheng, X.; Li, J.; Li, B.; Li, L.; Liu, D.; Lu, Y. Temporal Context Mining for Learned Video Compression. *IEEE Trans. Multimed.* **2023**, *25*, 7311–7322. [CrossRef]

19. Li, J.; Li, B.; Lu, Y. Hybrid Spatial-Temporal Entropy Modelling for Neural Video Compression. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 1503–1511.
20. Li, J.; Li, B.; Lu, Y. Neural video compression with diverse contexts. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 22616–22626.
21. Li, J.; Li, B.; Lu, Y. Neural video compression with feature modulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 26099–26108.
22. Jin, I.; Xia, F.; Ding, F.; Zhang, X.; Liu, M.; Zhao, Y.; Lin, W.; Meng, L. Customizable ROI-Based Deep Image Compression. *arXiv* **2025**, arXiv:2507.00373. [CrossRef]
23. He, D.; Zheng, Y.; Sun, B.; Wang, Y.; Qin, H. Checkerboard context model for efficient learned image compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 14771–14780.
24. Ballé, J.; Minnen, D.; Singh, S.; Hwang, S.J.; Johnston, N. Variational image compression with a scale hyperprior. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–23.
25. x265 HEVC Encoder/h.265 Video Codec. Available online: https://www.x265.org/ (accessed on 1 July 2025).
26. Agustsson, E.; Minnen, D.; Johnston, N.; Ballé, J.; Hwang, S.J.; Toderici, G. Scale-space flow for end-to-end optimized video compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 8500–8509.
27. Hu, Z.; Lu, G.; Guo, J.; Liu, S.; Jiang, W.; Xu, D. Coarse-to-fine deep video coding with hyperprior-guided mode prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5911–5920.
28. Rippel, O.; Anderson, A.G.; Tatwawadi, K.; Nair, S.; Lytle, C.; Bourdev, L. ELF-VC: Efficient learned flexible-rate video coding. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 14459–14468.
29. Lin, J.; Liu, D.; Li, H.; Wu, F. M-LVC: Multiple Frames Prediction for Learned Video Compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 3543–3551.
30. Brand, F.; Seiler, J.; Kaup, A. On benefits and challenges of conditional interframe video coding in light of information theory. In Proceedings of the 2022 Picture Coding Symposium (PCS), San Jose, CA, USA, 7–9 December 2022; pp. 289–293.
31. Mentzer, F.; Toderici, G.; Minnen, D.; Hwang, S.J.; Caelles, S.; Lucic, M.; Agustsson, E. VCT: A video compression transformer. In Proceedings of the 36th Conference on Neural Information Processing Systems (NIPS), New Orleans, LA, USA, 10–16 December 2023; pp. 13091–13103.
32. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
33. Lu, M.; Duan, Z.; Zhu, F.; Ma, Z. Deep hierarchical video compression. In Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence (AAAI), Vancouver, BC, Canada, 20–27 February 2024; pp. 8859–8867.
34. Jia, Z.; Li, B.; Li, J.; Xie, W.; Qi, L.; Li, H.; Lu, Y. Towards Practical Real-Time Neural Video Compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 10–17 June 2025; pp. 12543–12552.
35. Jiang, W.; Li, J.; Zhang, K.; Zhang, L. Ecvc: Exploiting non-local correlations in multiple frames for contextual video compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 10–17 June 2025; pp. 7331–7341.
36. Mercat, A.; Viitanen, M.; Vanne, J. UVG dataset: 50/120 fps 4K sequences for video codec analysis and development. In Proceedings of the 11th ACM Multimedia Systems Conference, Istanbul, Turkey, 8–11 June 2020; pp. 297–302.
37. Sheng, X.; Li, L.; Liu, D.; Wang, S. Bi-directional deep contextual video compression. *IEEE Trans. Multimed.* **2025**, 1–15. [CrossRef]
38. Zhai, Y.; Tang, L.; Jiang, W.; Yang, J.; Wang, R. L-LBVC: Long-Term Motion Estimation and Prediction for Learned Bi-Directional Video Compression. In Proceedings of the IEEE Data Compression Conference (DCC), Snowbird, UT, USA, 19–21 March 2025; pp. 53–62.
39. Alexandre, D.; Hang, H.M.; Peng, W.H. Hierarchical B-frame video coding using two-layer CANF without motion coding. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 10249–10258.
40. Ranjan, A.; Black, M.J. Optical Flow Estimation Using a Spatial Pyramid Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2720–2729.
41. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML* **2013**, *30*, 3.

42. Shi, W.; Caballero, J.; Huszar, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
43. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
44. Xue, T.; Chen, B.; Wu, J.; Wei, D.; Freeman, W.T. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.* **2019**, *127*, 1106–1125. [CrossRef]
45. Ma, D.; Zhang, F.; Bull, D.R. BVI-DVC: A Training Database for Deep Video Compression. *IEEE Trans. Multimed.* **2022**, *24*, 3847–3858. [CrossRef]
46. Bossen, F. Common test conditions and software reference configurations. In Proceedings of the 3rd. JCT-VC Meeting, Guangzhou, China, 7–15 October 2013; Volume 12.
47. Wang, H.; Gan, W.; Hu, S.; Lin, J.Y.; Jin, L.; Song, L.; Wang, P.; Katsavounidis, I.; Aaron, A.; Kuo, C.C.J. MCL-JCV: A JND-based H.264/AVC video quality assessment dataset. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 1509–1513.
48. Joint Video Expoloration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG. HEVC Reference Software. Version 18.0. Available online: https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tree/HM-18.0?ref_type=tags (accessed on 17 January 2025).
49. FFmpeg. Version 7.0. Available online: https://github.com/FFmpeg/FFmpeg/tree/release/7.0 (accessed on 4 September 2024).
50. Joint Video Expoloration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG. VVC Reference Software. Version 23.4. Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-23.4?ref_type=tags (accessed on 21 October 2024).
51. Bjøntegaard, G. Calculation of average PSNR differences between RDcurves. In Proceedings of the Thirteenth Meeting, Austin, TX, USA, 2–4 April 2001; Tech. Rep., VCEG-M33, ITU-T SG16/Q6.
52. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
53. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. In Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Advanced ANN Architecture for the CTU Partitioning in All Intra HEVC

Jakub Kwaśniak, Mateusz Majtka, Mateusz Lorkiewicz, Tomasz Grajek and Krzysztof Klimaszewski *

Institute of Multimedia Telecommunications, Poznan University of Technology, pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland; tomasz.grajek@put.poznan.pl (T.G.)
* Correspondence: krzysztof.klimaszewski@put.poznan.pl

Abstract

Due to the growing complexity of video encoders, the optimization of the parameters of the encoding process is becoming an important issue. In recent years, this has become an important field of application of neural networks. Artificial neural networks in video encoders are used to accelerate the video encoder operation. This paper demonstrates the use of different ResNet- and DenseNet-type architectures to accelerate the CTU partitioning algorithm in HEVC in All Intra mode. The paper demonstrates the results of an exhaustive evaluation of different proposed architectures, considering compression efficiency, network size, and encoding time reduction. Multiple pros and cons of the proposed architectures are presented in the Conclusions, considering various limitations that may be important for a given application, like hardware-constrained sensor networks or standalone small devices operating with images and videos.

Keywords: artificial neural networks; HEVC; ResNet; DenseNet; video encoding

1. Introduction

One of the most recent video coding technologies is High Efficiency Video Coding (HEVC) [1]. This video coding standard is used, among others, in digital TV broadcasting. HEVC offers improved coding performance compared to previous standards, especially for high-resolution videos. Its efficiency in terms of providing a high compression ratio and keeping the quality of the reconstructed images at acceptable levels made it possible to use also use HEVC in the compression of static images [2]. Although HEVC was standardized in 2013, the work on improving its operation is still ongoing. Especially in hardware-constrained or power-constrained applications, the older coding standards still find their use due to lower complexity and the remaining potential for optimizing their performance. Even the preceding standard, Advanced Video Coding (AVC), is still a subject of ongoing research [3]. The more recent Versatile Video Coding (VVC) standard is significantly more complex, and its adaptation to constrained environments is more problematic. The research on optimizing its performance is also ongoing, as evidenced by the recent papers [4,5].

In our work, we focus on HEVC mostly due to the fact that its intra-frame encoding quality is sufficient for encoding still images while being significantly less complex than VVC. Other researchers have also studied possible ways to improve the performance of HEVC. In [6], the authors focus on optimizing the rate control of HEVC, while our work focuses on a more time-consuming process of choosing the CTU divisions.

One of the properties of HEVC that makes it possible to obtain a high compression ratio is the use of advanced hierarchical division of the image into blocks. At first, the whole image or video frame is divided into blocks of the same size (usually 64 × 64 pixels) called a Coding Tree Unit (CTU). CTUs can be further divided using a quadtree structure into Coding Units (CUs) down to the size of 8 × 8 pixels. Those Coding Units undergo the coding process. The process of establishing the way the given CTU is divided is usually called partitioning. The complex structure of divisions makes it challenging to choose the optimal one. The division of CTUs of an image or video frame into CUs is the most time-consuming part of the encoding [7]. The example structure of divisions for a single CTU is presented in Figure 1.

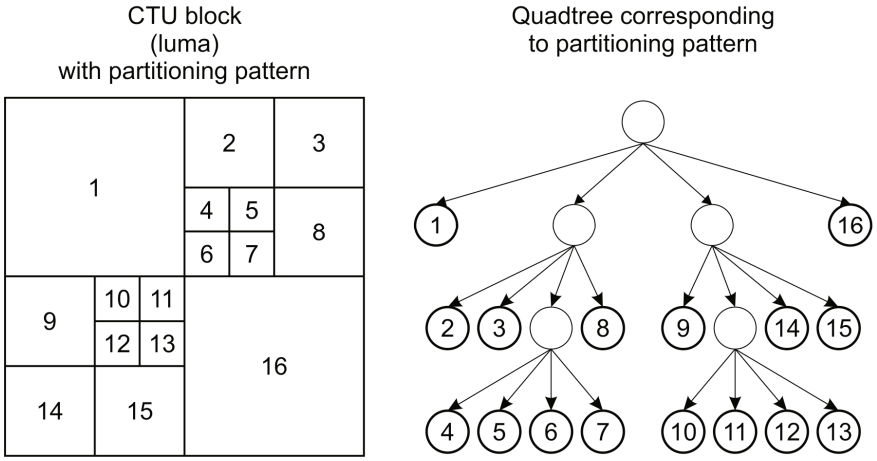


Figure 1. An example of a CTU division. The divided CTU is shown on the left and the corresponding quadtree on the right. Each leaf on the right can have its own division flag that signals whether it is further divided or not.

The division example shown in Figure 1 can be signaled as a set of division flags. In the implementation of encoders and decoders, the same information can be stored in a division matrix—a rectangular matrix of the size of 16 × 16 cells (for the CTU size of 64 × 64) that includes the information of the level of division for a given area of CTU. An example of such a matrix for the same case as shown in Figure 1 is presented in Figure 2.

1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	3	3	3	3	2	2	2	2
1	1	1	1	1	1	1	1	3	3	3	3	2	2	2	2
1	1	1	1	1	1	1	1	3	3	3	3	2	2	2	2
1	1	1	1	1	1	1	1	3	3	3	3	2	2	2	2
2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1
2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1
2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1
2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1

Figure 2. The division matrix for the same division as shown in Figure 1. The numbers indicate the level of the CTU division.

The computational complexity of CTU processing is a widely studied problem among researchers. In most cases, the goal is to speed up the process while maintaining the

compression ratio as high as possible. The baseline of this research is usually the reference implementation of an HEVC encoder, called then HEVC Test Model (HM) [8]. Different methods are employed to perform a fast and efficient partitioning of a CTU. One way to approach the problem of speeding up the partitioning process is to employ refined algorithms to perform partitioning based on different properties of the compressed image, such as the texture [9], the statistical properties [10], or the heuristic properties [11]. Those methods are usually used to recursively predict the division flags on consecutive levels of the quadtree. Another approach that has been explored in recent years is the use of artificial neural networks (ANNs). For this approach, simple convolutional network structures such as AlexNet can be used to predict the division flags [12,13] or to predict those flags for the entire CTU [14]. Also, more advanced network architectures, such as ResNet [11] or DenseNet [12], are used to recursively predict the division flags. The further development of convolutional networks used to predict the entire division matrix for a given CTU is also proposed in the literature [15].

This paper presents the further development of the networks based on ResNet [11] and DenseNet [12] architectures for efficiently predicting the partitioning of the entire CTU in an HEVC encoder. The possible tradeoffs to be made when choosing the network structure are demonstrated. Experimental results for diverse modifications of the network structures are presented, along with a comparison with a fully convolutional reference network. The focus is on All Intra coding, where the encoder compresses each video frame separately without motion compensation prediction. All Intra mode can easily be used to encode both still images and videos, providing low latency in encoding and decoding.

2. Materials and Methods

2.1. Experimental Setup

The experiment procedure is presented in Figure 3. In experiments, the modified HEVC encoder [15] is used. This software allows the use of pre-trained ANN models as long as the input and output formats are compatible. Therefore, the input to the proposed ANNs is luminance samples associated with the currently processed CTU block: an image fragment of 64 × 64 pixels. Further, the output is a 4 × 4 × 4 tensor containing the estimated probabilities of a given division depth for the considered block of 64 × 64 pixels constituting a CTU. Such dimensions of the output tensor are determined by the most typical configuration for HEVC. Since the CTU can be divided down to CU blocks of 8 × 8 pixels, the division matrix (Figure 2) can be reduced from size 16 × 16 to 4 × 4. This constitutes the first two dimensions of the output tensor. The size of the last dimension is tied to the maximal depth of the quaternary tree. Since the CTU can be divided down to CU blocks of 8 × 8 pixels, there are four possible division depths for a 64 × 64 pixel CTU (no division at all = 0, division into 32 × 32 pixels blocks = 1, division into 16 × 16 pixels blocks = 2, and division into 8 × 8 pixels blocks = 3).

A number of different networks were prepared based on both ResNet and DenseNet, varying in the details of the structure. A more specific description, with a list of proposed network variations, is presented in Section 2.2.

For training purposes, a dedicated training dataset was prepared, using the DIV2k image dataset [16]. The DIV2k consists of training (800 images) and validation (100 images) subsets. To prepare a dedicated training dataset, the images from the DIV2k dataset were encoded using the HM (version 16.23) [8] in All Intra mode. Then, training samples were composed by pairing a CTU luminance sample (64 × 64 pixels) with a partitioning chosen by the HM. This results in a dedicated training dataset comprising 589,589 samples in the training subset and 66,650 in the validation subset.

It is important to mention that the partitioning for a certain CTU block differs depending on the quantization parameter (QP) value. Therefore, a separate training dataset was prepared for each QP value indicated in the Common Test Conditions for HEVC (CTC) [17]. This means that a separate ANN model is trained for each QP value and then used in evaluation.

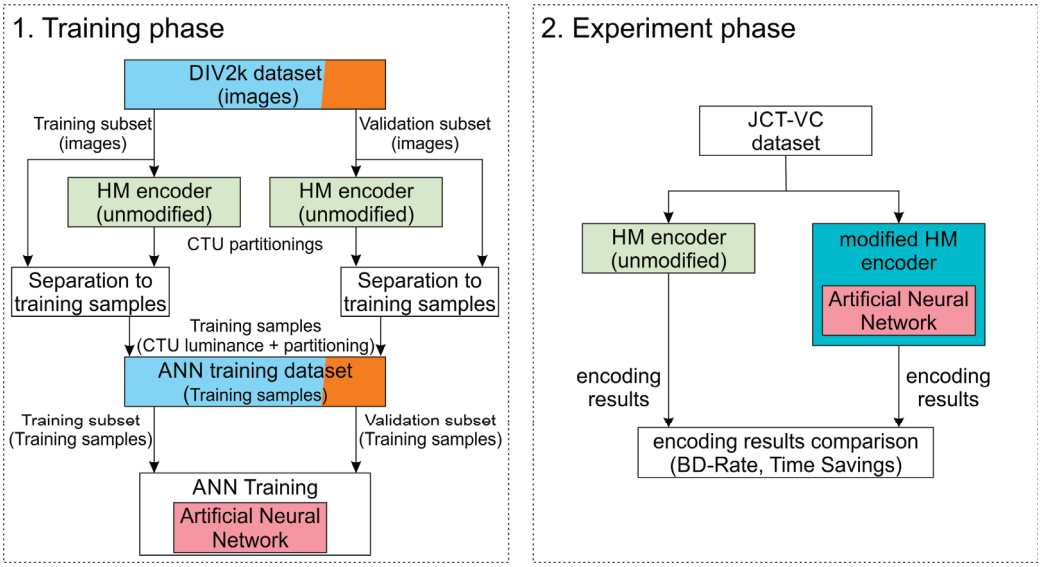


Figure 3. The experiment procedure. Full description is given in the text.

All networks were trained for 50 epochs, restarting the optimizer state every five epochs. Adaptive Moment Estimation (ADAM) [18] was used as an optimizer, and the loss function was cross-entropy. The batch size was 64, and training samples were shuffled every epoch. The training was performed using the Tensorflow [19] framework (version 2.5) with GPU acceleration (Python 3.8, CUDA 11.2, cuDNN 8.1, nVidia series 16, 20, and 30). The final model was converted to the corresponding model in LibTorch [20] (version 1.8.1 for CPU).

Once trained, the networks were used to encode the final evaluation with the modified HEVC encoder [15]. The final evaluation was performed with the Joint Collaborative Team on Video Coding (JCT-VC) dataset, defined in CTC [17]. Table 1 presents sequences in JCT-VC. The modified HEVC encoder [15] is based on the HM in version 16.23; thus, this HM version was used to produce reference results. The appropriate CTC for the All Intra scenario was applied without any changes. This means that encoding was performed for four QP values: 22, 27, 32, and 37.

One should mention that a modified HEVC encoder running ANNs uses only CPU, which is limited to a single logical core and a single thread. Such an approach ensures a fair comparison with the HM (single-threaded implementation) encoder [8]. To estimate the performance of the ANN, the results from the modified HM with the tested ANN are compared with the results of the HM for the same sequence and conditions. The following parameters were collected during the final evaluation: the number of additions and multiplications performed in millions of operations, the number of network parameters in thousands of parameters, the changes in the stream produced by the encoder, expressed as a Bjøntegaard delta [21], and finally the encoding time reduction, expressed in the percent of the reference encoder [8] encoding time. The final evaluation was performed on a single platform consisting of an R9 7950X processor with 96 GB of DDR5 RAM and NVME SSD for data storage under the control of Ubuntu in version 24.04.

Table 1. The list of the sequences used for evaluation.

JCT-VC Class	Sequence Name	Resolution	Number of Frames	Framerate	Bit Depth
A	NebutaFestival	2560 × 1600	300	60	10
	PeopleOnStreet	2560 × 1600	150	30	8
	SteamLocomotiveTrain	2560 × 1600	300	60	10
	Traffic	2560 × 1600	150	30	8
B	BQTerrace	1920 × 1080	600	60	8
	BasketballDrive	1920 × 1080	500	50	8
	Cactus	1920 × 1080	500	50	8
	Kimono1	1920 × 1080	240	24	8
	ParkScene	1920 × 1080	240	24	8
C	BasketballDrill	832 × 480	500	50	8
	BQMall	832 × 480	600	60	8
	PartyScene	832 × 480	500	50	8
	RaceHorses	832 × 480	300	30	8
D	BasketballPass	416 × 240	500	50	8
	BlowingBubbles	416 × 240	500	50	8
	BQSquare	416 × 240	600	60	8
	RaceHorsesLow	416 × 240	300	30	8
E	FourPeople	1280 × 720	600	60	8
	Johnny	1280 × 720	600	60	8
	KristenAndSara	1280 × 720	600	60	8
	FourPeople	1280 × 720	600	60	8

Since the raw output of the neural network may not be compatible with the HEVC syntax, the output tensor must be adjusted to comply with the HEVC standard requirements. Therefore, during the operation of the modified HEVC encoder, after the ANN processing, it is ensured that the division map is a valid CTU quadtree division. A heuristic algorithm is used to adjust the values. First, the result of the ArgMax function of the output tensor (for the last tensor dimension) is calculated, resulting in a 4×4 matrix with division depth values. Then, this matrix is adjusted in the following manner. If the number of division depth values being equal to 0 is greater than 8, then the whole CTU division depth is set to 0. Following, the smaller regions (according to the quaternary tree) are considered with adjusted thresholds for depth values of 1 and 2. At this stage, the matrix is also converted to a 16×16 matrix as used internally in the HEVC encoder by multiplicating the values contained in a 4×4 matrix.

2.2. Examined Network Structures

This research examines two ANN architectures, ResNet [22] and DenseNet [23]. These architectures are more advanced variants of fully convolutional networks, which utilize a shortcut connection or connections between layers in the network. In this research, tested ANNs are composed of specific blocks, according to a given architecture. To ensure the proper size of the output tensor, ANNs are composed of at least four such specific blocks, each paired with a MaxPool layer (pool size and stride both equal 2). Some DenseNet ANNs use only three blocks; thus, the pooling size in the last pooling layer is adjusted accordingly. If not specified in the modification description, only one specific block is applied before the pooling layer. Another restriction is the number of feature maps outputted from the last convolution layer in the ANN, which must be four. Lastly, the last layer in the ANN is

always a Softmax. In the proposed network, convolution filters always have a size of 3×3, and appropriate padding is applied (a copy of edge samples).

The first ANN architecture examined in this research is ResNet [24]. A schematic diagram of the ResNet architectures used is presented in Figure 4. The characteristic feature of this architecture is the residual connection. This residual connection conveys feature maps to the end of the block, omitting the main processing track in the block. Within the residual connection, the feature maps can be processed by multiple convolutional layers (or none) [25]. In this research, a residual connection contains a single convolution layer with Batch Normalization. The output of a block is the sum of the feature maps from the main processing track and residual connection, processed by activation (PReLU [26]).

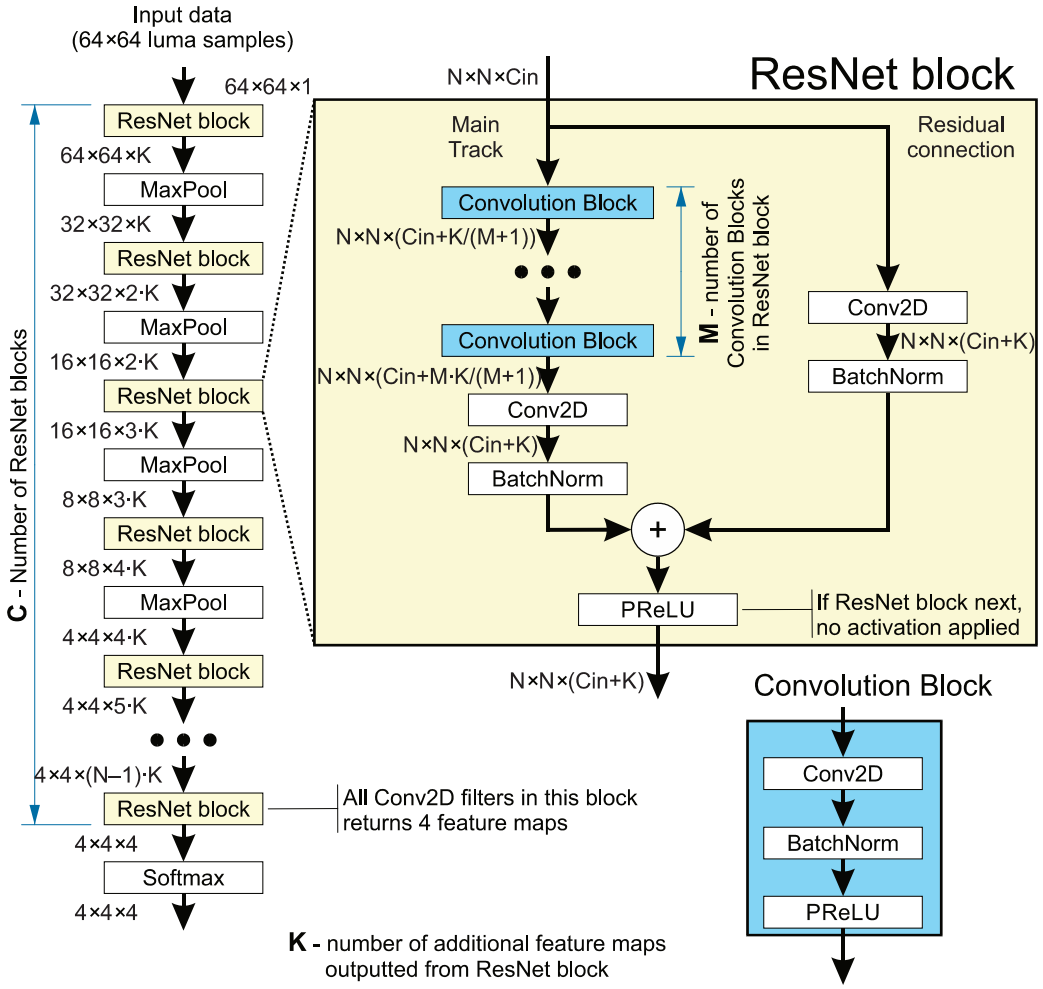


Figure 4. Base architecture of the ResNet network used. Please note that for each "ResNet block" the Conv2D blocks will have a different number of filters to provide the correct number of output layers, as shown on the left.

In the description of the ResNet architectures, one will refer to the term Convolution Block as an element of the network. In this paper, this refers to three layers: 2D Convolution, Batch Normalization, and PReLU, connected in the presented order. See the blue inset in Figure 4.

A number of variants were examined in this research. Table 2 presents the list of variants, whose results are presented and discussed in this paper. The number of Convolution Blocks refers to Convolution Blocks used in the main track of the ResNet block.

Table 2. Description of the examined architectures of ResNet-based networks. The number of Convolution Blocks refers to the Convolution Block in the main track of the ResNet block.

Network Name	Description	Number of ResNet Blocks (C)	Convolution Blocks in ResNet Block (M)	Number of Additional Feature Maps from ResNet Block (K)
Mod0	Base architecture	7	2	6
Mod1	AvgPool2D as a pooling layer	7	2	6
Mod2	Two ResNet blocks before each MaxPool	8	2	6
Mod3	Single Convolution Block in the ResNet block, more feature maps added by the ResNet block	7	1	8
Mod4	Four Convolution Blocks in the ResNet block, two less ResNet blocks, and more feature maps added by the ResNet block	5	4	10
Mod5	Three Convolution Blocks in the ResNet block, more feature maps added by the ResNet block	7	3	6
Mod6	Three ResNet blocks before MaxPool	12	2	6
Mod7	ReLU as activation	7	2	6
Mod8	Additional ResNet block	8	2	6
Mod9	One ResNet block less	6	2	6
Mod10	More feature maps added by the ResNet block	7	2	12

The second network architecture examined in the research was DenseNet [27]. A schematic diagram of the DenseNet architecture used is presented in Figure 5. The distinctive part of these networks is the dense connection applied between all DenseNet layers with the DenseNet block. This means that consecutive layers in the blocks jointly process all feature maps, both from the input of the block and all previous layers. In this research, the dense connections are achieved by outputting from the layer the concatenation of generated and inputted feature maps of the given layer. The block always ends with a single Conv2D, which processes accumulated feature maps and reduces their number. In this research, the first block always outputs eight feature maps, while the rest output four feature maps.

One should mention that in some architectures, only three DenseNet blocks, paired with pooling layers, are used. Therefore, the pool size and stride are set to four in the last pooling layer to maintain the desired output tensor size.

In the presented research, two types of DenseNet layers were examined. In the first one, referred to as Default, the order of layers in the main track is BatchNorm, PReLU, and Conv2D. In the second one, referred to as Shifted, the order of layers in the main track is Conv2D, BatchNorm, and PReLU. Depending on the type of DenseNet layers used, the number of convolutions in consecutive layers for a given DenseNet block differs. In the case of Default DenseNet layers, the number of convolution filters in consecutive DenseNet layers is 4, 6, 8, 6, 4, 4. When Shifted DenseNet Layers are used, the number of convolution filters in consecutive DenseNet layers is 4, 6, 8, 8, 6, 4. One should mention that when fewer DenseNet layers in the DenseNet block are applied, unnecessary layers are simply removed.

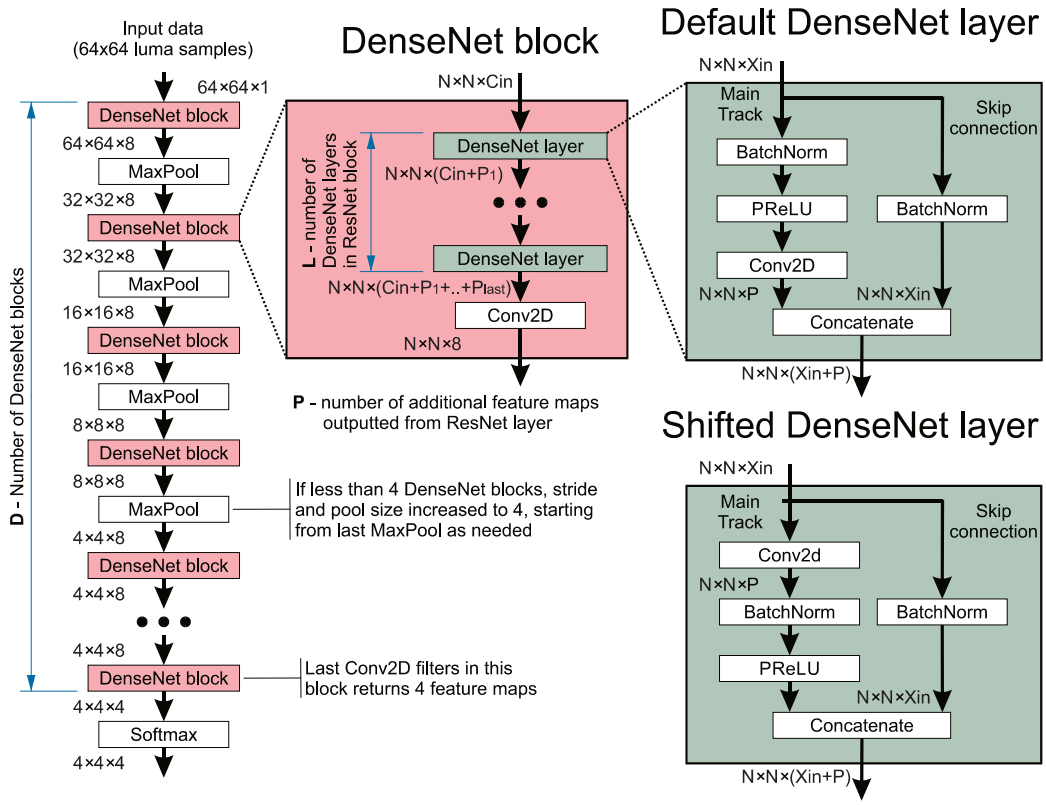


Figure 5. The base architecture of the DenseNet network used. Please note that for "DenseNet layer" and "DenseNet block", the Conv2D blocks will have a different number of filters, and the MaxPool layer will have a different size to provide the correct number of output layers, as shown on the left.

Table 3 presents the list of variants whose results are presented and discussed in this paper.

Table 3. Description of the examined architectures of DenseNet-based networks.

Network Name	Description	Number of DenseNet Blocks (D)	Number of DenseNet Layers in Each DenseNet Block (L)	Applied Dense Layer Type
Mod0	Base architecture	3	6	Default
Mod1	ReLU as activation	3	6	Default
Mod2	AvgPool2D as a pooling layer	3	6	Default
Mod3	4 DenseNet blocks, with 4 DenseNet layers in each	4	4	Default
Mod4	Using Shifted DenseNet layers	3	6	Shifted
Mod5	Using 5 DenseNet blocks with 4 DenseNet layers each	5	4	Default
Mod6	Using 6 DenseNet blocks with 4 DenseNet layers each	6	4	Default

Table 3. *Cont.*

Network Name	Description	Number of DenseNet Blocks (D)	Number of DenseNet Layers in Each DenseNet Block (L)	Applied Dense Layer Type
Mod7	Using 3 DenseNet blocks with 5 Shifted DenseNet layers each	3	5	Shifted
Mod8	Using 4 DenseNet blocks with 5 Shifted DenseNet layers each	4	5	Shifted
Mod9	Using 5 DenseNet blocks with 5 Shifted DenseNet layers each	5	5	Shifted

3. Results

All the networks were trained on the DIV2K-based dataset and evaluated in a modified HEVC encoder [15] using the JCT-VC dataset [17], according to the description presented in Section 2.1. None of the sequences from the JCT-VC dataset [17] were used during the training process. In Table 4, we present the results of the experiments as a set of data gathered during the experiment. Please note that the lower the value for the BD-Rate, the better. For Time Saving, the higher the better. In this paper, only Time Savings for encoding are presented, as the decoding time did not change meaningfully.

Table 4. The results of the experiment.

Network	BD-Rate [21] [%]	Time Saving Compared to Original HM [8] [%]	Number of Additions and Multiplications (MAC) Operations (in Millions)	Number of Model Parameters (Weights) (in Thousands)
Full CNN [13]	1.87	56.08	6.76	91.600
ResNet Mod0	1.88	45.41	8.74	97.202
ResNet Mod1	1.81	45.74	8.74	97.202
ResNet Mod2	1.70	28.54	55.00	97.148
ResNet Mod3	1.87	51.27	11.13	128.188
ResNet Mod4	1.77	40.47	32.66	150.992
ResNet Mod5	1.81	43.22	19.44	212.878
ResNet Mod6	1.71	−7.46	204.08	583.968
ResNet Mod7	1.91	49.59	8.74	96.852
ResNet Mod8	1.88	47.67	9.62	152.960
ResNet Mod9	1.80	50.36	8.10	57.086
ResNet Mod10	1.77	44.28	24.49	160.262
DenseNet Mod0	1.84	32.00	40.49	23.170
DenseNet Mod1	1.96	36.08	40.49	22.820
DenseNet Mod2	1.87	32.48	40.49	23.170
DenseNet Mod3	1.90	39.11	26.06	18.330
DenseNet Mod4	1.84	28.19	41.16	24.540
DenseNet Mod5	1.92	25.57	44.75	22.380
DenseNet Mod6	1.82	12.11	60.11	26.420
DenseNet Mod7	1.85	34.86	33.38	19.890
DenseNet Mod8	1.84	31.61	33.73	25.670
DenseNet Mod9	1.79	17.49	59.15	31.450

One should mention that Table 4 includes results for a fully convolutional ANN delivered by the authors of the modified HM encoder [15]. This ANN was evaluated using the exact same procedure as the proposed ANNs. As it is one of the most popular and effective solutions found in the literature and represents a much simpler convolutional architecture, it is an excellent reference point for the ANNs proposed in this paper.

The fully convolutional network [13] used as a reference method has a very similar architecture to the proposed networks. It consists of seven blocks connected in series. The first five are Convolution Blocks (exactly the same as those used in the ResNet architecture), of which the first four are paired with MaxPool. Then, feature maps are split into four equal parts along the vertical and horizontal axes and processed with four independent Convolution Blocks. Further, feature maps are concatenated to restore the previous size and processed with Convolution Blocks but with convolution filters of size 1×1. Lastly, Softmax is applied to produce the output tensor.

One should mention that more ANN-based CTU partitioning algorithms are available in the literature. Worth mentioning are the methods presented by Feng [28] (BD-Rate: 1.75%, Time Saving: 60.35%), Xu [29] (BD-Rate: 2.25%, Time Saving: 61.85%), and Wang [30] (BD-Rate: 1.86%, Time Saving: 61.23%). However, certain aspects of the evaluation of results led to not including these results in the comparison. Firstly, models and encoder implementations were not available. The difference in used hardware for evaluation is an important factor that may disrupt the comparison. Secondly, older versions of the HM software used for evaluation (16.20 for Feng [28] and 16.5 for Xu [29] and Wang [30]) differ in partitioning decisions and computing time. Next, for ANN training in the mentioned methods, different training datasets were used. This aspect is crucial, as in this research focus is put strictly on ANN architecture. Lastly, as the scope of this research is on embedded devices, some aspects of ANN in these methods, like number of weights (for Xu [29] and Wang [30]—AlexNet-like architecture) or the number of MAC operations (for Feng [28]), exclude the use of these models in embedded solutions.

For ease of comparison, the results are presented in the form of graphs. As the most illustrative ones, the following ones are presented: Time Saving as a function of number of MAC operations (Figure 6), Time Saving as a function of BD-Rate (Figure 7), Time Saving as a function of number of weights (Figure 8), BD-Rate as a function of number of weights (Figure 9), and BD-Rate as a function of number of MAC operations (Figure 10).

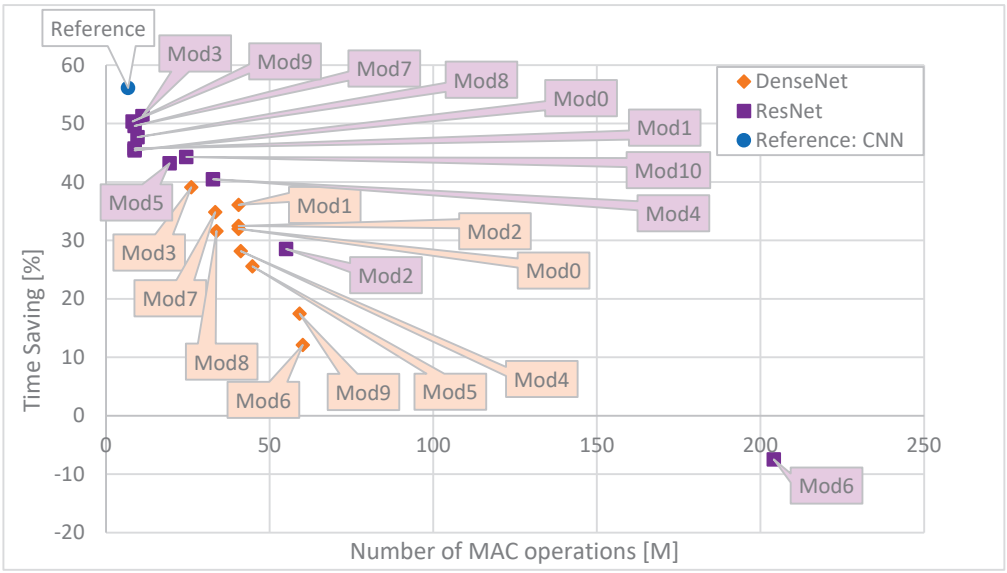


Figure 6. Correspondence between Time Saving and the number of operations performed for each network. The best ones are close to the upper left corner.

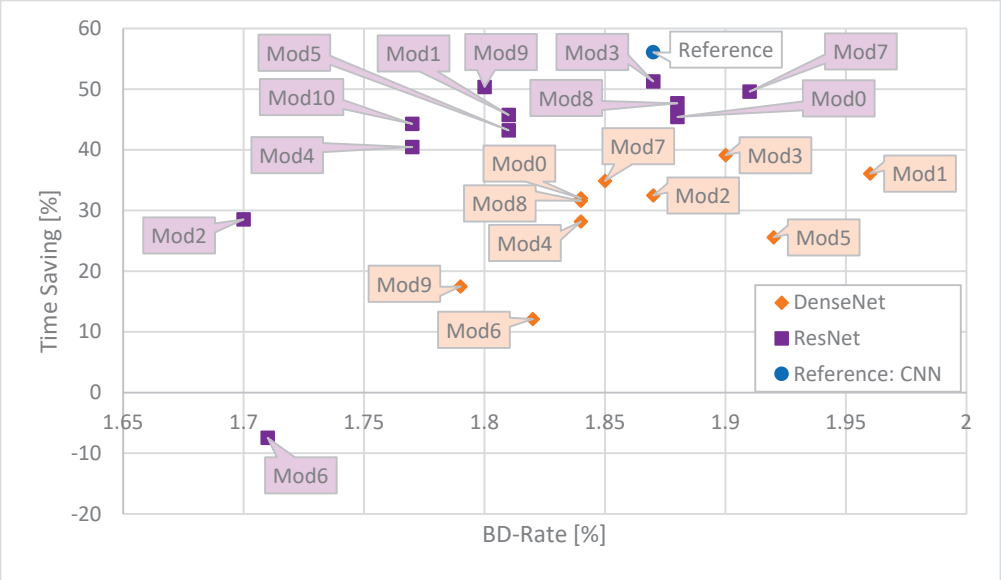


Figure 7. Correspondence between Time Saving and BD-Rate of the resulting bitstream for each network. The best ones are close to the upper left corner.

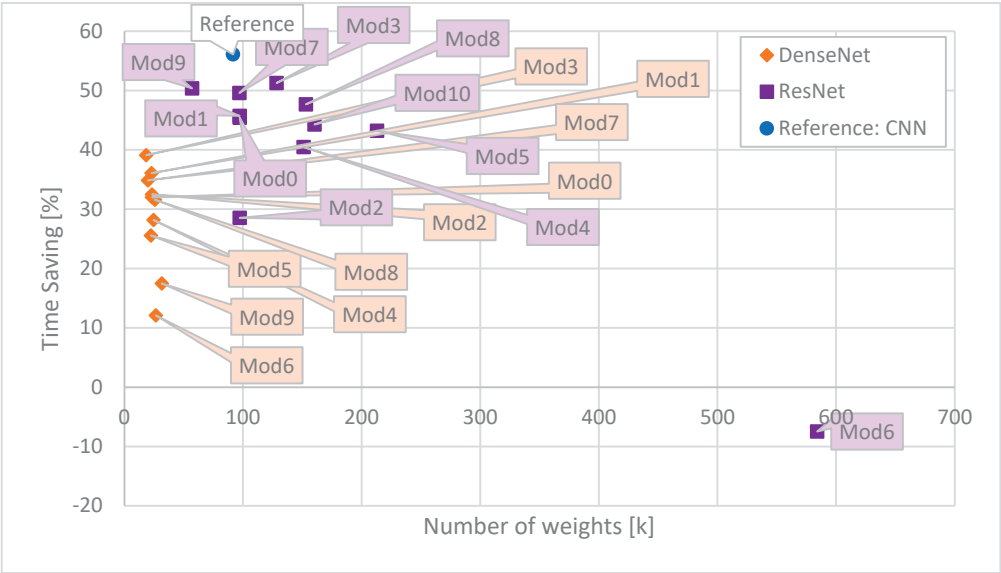


Figure 8. Correspondence between Time Saving and the number of weights for each network. The best ones are close to the upper left corner.

Regardless of the proposed variant of the ResNet or DenseNet, it can be seen (Figure 6) that the full convolutional network provides the best Time Saving with the fewest number of operations. The ResNet architecture in most variants requires about 30% more MAC operations; however, the Time Savings are smaller by 5 to 10 p.p. compared to a full convolutional network. Despite a very similar number of MAC operations, the results of Time Savings for ResNet Mod0 and Mod9 vary by 5 p.p. The major difference between those two variants is the number of ResNet blocks—Mod9 has one less. On the other hand, the results for Mod10 are very similar to Mod0 (1 p.p. difference) despite a much bigger number of MAC operations. This means that the number of layers is much more impactful on ANN computing time than the number of convolution filters. In our experiments, DenseNet architectures are significantly slower than ResNet-based networks. Despite fewer MAC

operations in DenseNet Mod3, the ResNet Mod 4 provides better Time Saving. This means that a greater number of skip connections caused a significant latency in computations, thus deteriorating the Time Savings results. Therefore, the number of residual/dense blocks impacts the Time Savings differently. Considering ResNet, the most complex modification is increasing the number of blocks before the pooling layers. As more blocks process feature maps of bigger dimensions, both the encoding time and number of MAC operations rapidly increase. Having the DenseNet, each additional block significantly increases the number of MACs, which results in ~5 p.p. lower Time Savings per additional block.

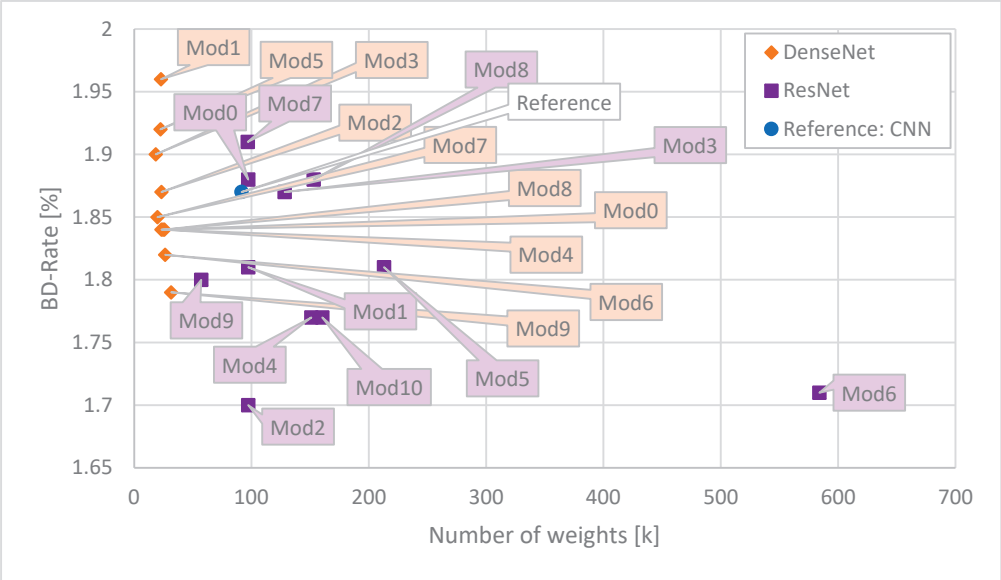


Figure 9. Correspondence between BD-Rate and number of weights for each network. The best ones are close to the lower left corner.

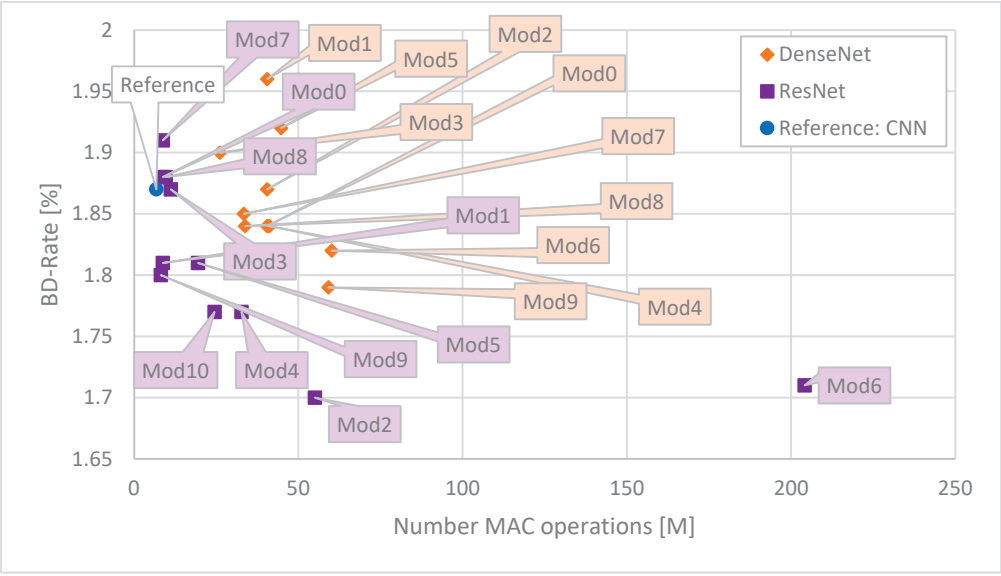


Figure 10. Correspondence between BD-Rate and number of weights for each network. The best ones are close to the lower left corner.

A broader perspective on the performance of proposed networks is shown when BD-Rate is considered. As demonstrated in Figure 7, the bitstream produced by the encoder is

slightly bigger than the unmodified HM encoder, despite the used network. The two most complex (number of MAC operations) variants of ResNet, Mod 2 and Mod 6, obtained the best results, around 1.7%, which outperforms the reference CNN by ~0.2 p.p. However, the computational complexity of the network is not directly related to the bitstream produced by the encoder. Among the least complex ResNet architectures, in terms of BD-Rate, some of them perform better (e.g., Mod9) and some worse (e.g., Mod 7) compared to the reference CNN architecture. Considering DenseNet architectures, most of the proposed networks outperform the reference CNN, with results comparable to most ResNet architectures. Among the two tested DenseLayer types, the Shifted is more effective. However, ResNet (purple) Mod9 can outperform all but one (orange Mod9) DenseNet network in terms of BD-Rate.

One should recall that both ResNet and DenseNet were tested with AveragePooling (Mod1 for ResNet and Mod 2 for DenseNet) and ReLU as activation (Mod 7 for ResNet and Mod 1 for DenseNet). These modifications did not change the complexity (number of MAC operations) or the size of the networks (number of parameters), as presented in Table 4. Considering Time Saving and BD-Rate, almost none of these modifications improved the performance of the network, as the results are worse compared to the non-modified versions of networks (Mod0) that use PReLU and MaxPool. The exception is ResNet with AveragePooling (Mod1), where BD-Rate is slightly better.

Yet another perspective is given when the number of weights is considered. As shown in Figure 8, the main advantage of the DenseNet architectures is the number of weights. The smallest ResNet architecture (Mod9) is slightly less than twice as big as the biggest DenseNet (Mod9). Moreover, the DenseNet (Mod9) has three times fewer weights than the reference CNN. Therefore, small size does not come with the encoding time, as most ResNet architectures provide higher Time Savings. In some modifications of ResNet, consecutive ResNet blocks produced more feature maps (Mod3, Mod4, and Mod10) as compensation for a lower number of ResNet blocks. One can observe that for such modifications, despite a significant increase in the number of weights, the Time Savings did not change compared to the Default ResNet (Mod0).

Unfortunately, the smallest network (DenseNet—orange—Mod3) causes the encoder to produce a bitstream larger than most of the competitors, as demonstrated in Figure 9. It is worth noting, though, that for hardware-constrained applications, the Time Saving provided by the DenseNet Mod3 network, accompanied by its smallest size, might be more important than the fact that the encoder produces a slightly larger bitstream. Overall, for DenseNet, the bigger the network, the better the BD-Rate. Different observations are made for the ResNet architectures. The BD-Rate result is more related to the block and later setup (Mod2, Mod 4 and Mod10) rather than overall network size (Mod 5 and Mod 6).

When the more important factor is the number of operations and the size of the network (number of weights) is less of an issue, ResNet-based networks will be a better choice (like purple Mod9). This is evidenced in Figure 10. DenseNet architectures provide slightly higher Time Saving with less than half of the number of operations than the best DenseNet (orange Mod3), causing a smaller output bitstream increase. DenseNet Mod3 provides the most Time Saving and is the smallest network among all the DenseNet-based networks in this study, requiring the least MAC operations to compute. On the other hand, its shortcoming is the production of a larger bitstream than most of the other networks in this study.

4. Conclusions

Artificial neural networks provide a feasible method of speeding up the compression process of an HEVC coder. In the paper, we presented a comparison of the performance of

different artificial neural networks used to predict the divisions in the CTU block for an HEVC encoder in All Intra mode. The results obtained during the presented study show that there is a noticeable difference between the network architectures, both in terms of their performance, size, and complexity in the said application. The studied HEVC All Intra compression mode is especially useful in applications where high-quality images are compressed. The HEVC coder proves to be an effective tool for still image compression. Reducing the complexity of the encoding process through an efficient network predicting the CTU divisions is an important improvement to the encoding process.

However, for some applications, it is important to keep the size of the network as small as practically possible. Although the most obvious method to reduce the size of the whole network is to reduce the number of blocks in the network, our study proves that it is more beneficial to change the overall structure of the network. Although ResNet networks provide better performance than DenseNet networks, trying to reduce the size of the ResNet network will decrease its performance to a point where it is better to change the structure of the network to a slightly worse-performing DenseNet structure that provides a significant reduction in size, time complexity, and number of MAC operations. This is especially important for applications with strict memory and power constraints, like in sensor networks or similar applications.

In future work, we intend to extend the study to the performance improvement of interframe coding in HEVC as well as study the performance of the networks described in this paper in other available encoders.

Author Contributions: Conceptualization, M.L. and T.G.; methodology, M.L.; software, M.L., J.K. and M.M.; validation, M.L., T.G. and K.K.; writing—original draft preparation, K.K. and M.L.; writing—review and editing, K.K., M.L. and T.G.; visualization, K.K. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Higher Education, Poland.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADAM	Adaptive Moment Estimation optimizer
AVC	Advanced Video Coding
CPU	Central Processing Unit
CTU	Coding Tree Unit
CU	Coding Unit
GPU	Graphics Processing Unit
HEVC	High Efficiency Video Coding
HM	HEVC Test Model (reference implementation of an HEVC)
JCT-VC	Joint Collaborative Team on Video Coding
QP	Quantization Parameter
VVC	Versatile Video Coding

References

1. *ISO/IEC IS 23008-2*; Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding. International Organization for Standardization (ISO): Geneva, Switzerland; International Electrotechnical Commission (IEC): Geneva, Switzerland, 2013.
2. Lainema, J.; Hannuksela, M.M.; Vadakital, V.K.M.; Aksu, E.B. HEVC still image coding and high efficiency image file format. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
3. Kau, L.-J.; Tseng, C.-K.; Lee, M.-X. Perception-Based H.264/AVC Video Coding for Resource-Constrained and Low-Bit-Rate Applications. *Sensors* **2025**, *25*, 4259. [CrossRef] [PubMed]
4. Park, D.; Moon, G.; Oh, B.T.; Kim, J.G. Coarse-to-Fine Network-Based Intra Prediction in Versatile Video Coding. *Sensors* **2023**, *23*, 9452. [CrossRef] [PubMed]
5. Filipe, J.N.; Tavora, L.M.; Faria, S.M.; Navarro, A.; Assuncao, P.A. Complexity reduction methods for Versatile Video Coding: A comparative review. *Digit. Signal Process.* **2025**, *160*, 105021. [CrossRef]
6. Chen, S.; Aramvith, S.; Miyanaga, Y. Learning-Based Rate Control for High Efficiency Video Coding. *Sensors* **2023**, *23*, 3607. [CrossRef] [PubMed]
7. Bossen, F.; Bross, B.; Suhring, K.; Flynn, D. HEVC Complexity and Implementation Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1685–1696. [CrossRef]
8. Fraunhofer Heinrich Hertz Institute. HEVC Reference Software. Available online: https://vcgit.hhi.fraunhofer.de/jvet/HM (accessed on 19 August 2025).
9. Zhang, Q.; Wang, Y.; Huang, L.; Jiang, B. Fast CU partition and intra mode decision method for H.266/VVC. *IEEE Access* **2020**, *8*, 117539–117550. [CrossRef]
10. Lu, Y.; Huang, X.; Liu, H.; Zhou, Y.; Yin, H.; Shen, L. Hierarchical classification for complexity reduction in HEVC inter coding. *IEEE Access* **2020**, *8*, 41690–41704. [CrossRef]
11. Kim, N.; Jeon, S.; Shim, H.J.; Jeon, B.; Lim, S.-C.; Ko, H. Adaptive keypoint-based CU depth decision for HEVC intra coding. In Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Nara, Japan, 1–3 June 2016.
12. Amna, M.; Imen, W.; Ezahra, S.F. LeNet5-Based approach for fast intra coding. In Proceedings of the 10th International Symposium on Signal, Image, Video and Communications, Saint-Etienne, France, 7–9 April 2021.
13. Feng, Z.; Liu, P.; Jia, K.; Duan, K. Fast Intra CTU Depth Decision for HEVC. *IEEE Access* **2018**, *6*, 45262–45269. [CrossRef]
14. Chen, Z.; Shi, J.; Li, W. Learned Fast HEVC Intra Coding. *IEEE Trans. Image Process.* **2020**, *29*, 5431–5446. [CrossRef] [PubMed]
15. Lorkiewicz, M.; Stankiewicz, O.; Domański, M.; Hang, H.M.; Peng, W.H. Complexity-Efficiency Control with ANN-Based CTU Partitioning for Video Encoding. *IEEE Access* **2024**, *12*, 102536–102551. [CrossRef]
16. Timofte, T.; Gu, A.; Wu, J. DIV2K Dataset: High-Quality Images for Super-Resolution and Image Enhancement. 2017. Available online: https://data.vision.ee.ethz.ch/cvl/DIV2K/ (accessed on 26 January 2025).
17. (JCT-VC). Joint Collaborative Team Video Coding, Common Test Conditions and Software Reference Configurations. In JCTVC-L1100, Incheon, Korea, 18–26 April 2013.
18. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
19. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016.
20. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
21. Bjøntegaard, G. Calculation of average PSNR differences between RD curves. In Proceedings of the ITU-T SG16/Q6, Document VCEG-M33, Austin, TX, USA, 2–4 April 2001.
22. Li, Y.; Li, L.; Zhuang, Z.; Fang, Y.; Yang, Y. ResNet Approach for Coding Unit Fast Splitting Decision of HEVC Intra Coding. In Proceedings of the IEEE 6th International Conference on Data Science in Cyberspace, Shenzhen, China, 9–11 October 2021.
23. Zhang, Q.; Guo, R.; Jiang, B.; Su, R. Fast CU Decision-Making Algorithm Based on DenseNet Network for VVC. *IEEE Access* **2021**, *9*, 119289–119297. [CrossRef]
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
25. Shafiq, M.; Gu, Z. Deep Residual Learning for Image Recognition: A Survey. *Appl. Sci.* **2022**, *12*, 8972. [CrossRef]

26. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
27. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
28. Feng, A.; Gao, C.; Li, L.; Liu, D.; Wu, F. Cnn-Based Depth Map Prediction for Fast Block Partitioning in HEVC Intra Coding. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021.
29. Xu, M.; Li, T.; Wang, Z.; Deng, X.; Yang, R.; Guan, Z. Reducing Complexity of HEVC: A Deep Learning Approach. *IEEE Trans. Image Process.* **2018**, *27*, 5044–5059. [CrossRef] [PubMed]
30. Wang, T.; Wei, G.; Li, H.; Bui, T.; Zeng, Q.; Wang, R. Fast CU Partition Algorithm for Intra Frame Coding Based on Joint Texture Classification and CNN. *Sensors* **2023**, *23*, 7923. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Review

Deep Learning for Image Watermarking: A Comprehensive Review and Analysis of Techniques, Challenges, and Applications

Marta Bistroń [1,*], Jacek M. Żurada [2] and Zbigniew Piotrowski [1]

[1] Institute of Communication Systems, Faculty of Electronics, Military University of Technology, 00-908 Warsaw, Poland; zbigniew.piotrowski@wat.edu.pl
[2] Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA; jacek.zurada@louisville.edu
* Correspondence: marta.bistron@wat.edu.pl

Highlights

What are the main findings?

- Deep learning-based watermarking methods (CNN, GAN, Transformers, and diffusion models) significantly outperform traditional spatial- and frequency-domain techniques in terms of robustness, transparency, and adaptability to modern attack types.
- Emerging architectures such as Vision Transformers, Swin Transformers, and diffusion models introduce new capabilities, notably higher resistance to generative and latent-space attacks, as well as increased watermark capacity.

What are the implications of the main findings?

- The rapid evolution of neural network architectures accelerates the development of watermarking systems capable of protecting digital content against increasingly sophisticated threats, including AI-generated manipulations.
- Future watermarking deployments will require optimized, scalable, and computationally efficient deep learning architectures to support real-time applications in cybersecurity, multimedia distribution, IoT systems, and content authenticity verification.

Abstract

The growing demand for digital content protection has significantly increased the importance of image watermarking, particularly in light of the rising vulnerability of multimedia content to unauthorized modifications. In recent years, research has increasingly focused on leveraging deep learning architectures to enhance watermarking performance, addressing challenges related to transparency, robustness, and payload capacity. Numerous deep learning-based watermarking methods have demonstrated superior effectiveness compared to traditional approaches, particularly those based on Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), Transformers, and diffusion models. This paper presents a comprehensive survey of recent developments in both conventional and deep learning-based image watermarking techniques. While traditional methods remain prevalent, deep learning approaches offer notable improvements in embedding and extraction efficiency, particularly when facing complex attacks, including those generated by advanced AI models. Applications in areas such as deepfake detection, cybersecurity, and Internet of Things (IoT) systems highlight the practical significance of these advancements. Despite substantial progress, challenges remain in achieving an optimal balance between invisibility, robustness, and capacity, particularly in high-resolution and real-time scenarios. This study concludes by outlining future research directions toward develop robust, scalable, and efficient deep

learning-based watermarking systems capable of addressing emerging threats in digital media environments.

Keywords: deep learning; digital watermarking; image watermarking; neural networks; robustness; watermarking algorithms

1. Introduction

The rapid development of internet and network technologies has led to the widespread digitization of everyday life [1]. Vast amounts of data (text, music, images, video) are processed and shared online on a daily basis. Digital data has become a fundamental resource underlying many social and economic activities. For this reason, one of the key challenges of the 21st century is to balance the protection of digital data with technological and economic advancement [2]. The main threats to digital content security include unauthorized copying and redistribution of content (multimedia piracy) [3–5], data manipulation and forgery [6], such as the creation of deepfakes [7], and the privacy-related abuses including unauthorized data usage.

To mitigate risks associated with digital data, various protection technologies have been developed, playing a pivotal role in ensuring data security. A classification of data-hiding techniques is presented in Figure 1. Among the mentioned methods, the most widely used are steganography, digital watermarking, and fingerprinting.

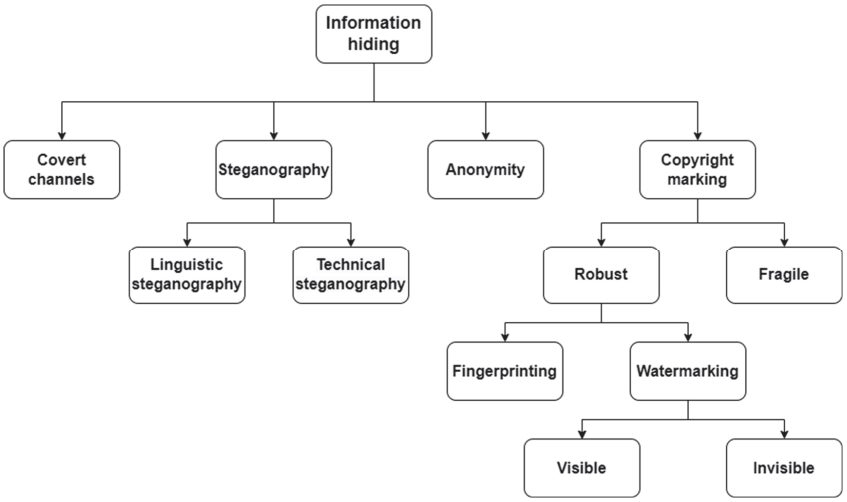


Figure 1. Classification of data hiding techniques based on [8].

Steganography involves the creation of hidden communication channels that enable the transmission of data in a manner imperceptible to third parties. The data is concealed within a carrier, which may include an image, video, audio, or text file [9]. Another form of steganography is network steganography, which focuses on embedding data within transmitted network packets or the communication mechanisms of protocols [10]. The primary objective of steganography is to conceal the very existence of communication. This distinguishes it from cryptography, which protects data through encryption but does not obscure the fact that communication takes place [11]. Digital watermarking focuses on embedding a watermark into a digital carrier, which can be either visible or invisible [12]. Unlike steganography, its purpose is not to conceal the existence of data but, most often, to identify the content owner. A similar technique is fingerprinting, which involves inserting unique markers into

https://doi.org/10.3390/s26020444

each copy of digital content. Both methods frequently rely on comparable technical solutions, with the difference lying in their purpose and scope of application. A watermark is typically embedded in the original content before distribution, whereas a fingerprint serves as a unique marker for each copy, enabling its identification and tracking [13]. Both techniques are often used in combination to provide more comprehensive protection of digital content.

Although steganography, watermarking, and fingerprinting play a crucial role in protecting digital data, they increasingly face sophisticated attacks and the growing complexity of multimedia content. Traditional algorithms often prove insufficient in addressing these issues. An increasingly popular solution is the application of deep learning algorithms, which can automatically learn patterns, adapt to various conditions, and demonstrate resilience to interference, making them effective tools for modern data-hiding techniques.

The subsequent sections of this article will focus exclusively on digital watermarking—its currently employed solutions and future development directions utilizing deep learning techniques. The key contributions of this work are as follows:

- A review and taxonomy of classical watermarking methods for images and video frames.
- A comprehensive overview and taxonomy of deep learning-based image watermarking techniques, highlighting their advantages, limitations, and potential application areas.
- Detailed comparisons of various deep learning architectures (CNNs, GANs, Transformers, and diffusion models) used in watermarking, with particular emphasis on their performance, robustness, and computational complexity.
- A review and comparison of key datasets used for training watermarking algorithms.
- An analysis of future research directions and practical challenges in areas such as deepfake detection, cybersecurity, and applications in IoT systems, with a special focus on the integration of deep learning methods into watermarking solutions.
- A discussion of dataset availability, training strategies, and the role of transparency metrics based on neural networks, as well as specialized robustness metrics tailored to assess the impact of generative and adversarial attacks, providing practical guidelines useful for real-world implementations.

In contrast to previous surveys [14,15] on deep learning-based watermarking, this work adopts a more detailed and practice-oriented perspective, focusing not only on the classification of methods but also on their architectural evolution and practical deployment. Most existing reviews concentrate on conventional deep learning architectures, such as convolutional neural networks (CNNs) and generative adversarial networks (GANs), commonly used in watermarking tasks.

In this review, we frame the development of watermarking techniques through the lens of architectural and design philosophy. Each new model represents an effort to overcome limitations of prior approaches and enhance the overall capacity and robustness of watermarking systems. Accordingly, we incorporate and analyze the latest techniques based on advanced architectures such as Vision Transformers (ViT, Swin Transformer) and diffusion models, which—as discussed in detail later in the paper—demonstrate clear advantages over traditional CNN- and GAN-based solutions, particularly in terms of robustness against attacks and embedding flexibility. Furthermore, this survey goes beyond theoretical discussion to address practical considerations related to real-world implementation. We examine challenges such as computational complexity, data availability, and the evaluation of watermarking performance in terms of both perceptual transparency and resilience to adversarial and generative attacks. This holistic approach aims not only to synthesize the current body of knowledge, but also to facilitate its application in the development and deployment of robust, scalable watermarking systems in practical engineering contexts.

https://doi.org/10.3390/s26020444

The remainder of this article is organized as follows. Section 2 presents the fundamentals of digital watermarking, providing a detailed description of the general workflow, key paradigms, and commonly used metrics. Section 3 reviews traditional watermarking methods, covering techniques in the spatial, frequency, and hybrid domains. Section 4 focuses on deep learning-based watermarking, discussing various architectures, including CNNs, GANs, Transformers, and diffusion models, with an in-depth comparison of their effectiveness and applications. Section 5 centers on datasets used in watermarking research, detailing available databases, their characteristics, and applications. Section 6 outlines future research directions, emphasizing architectural innovations and application-oriented challenges in areas such as deepfake detection, cybersecurity, and IoT. Section 7 concludes the article with a comprehensive summary, highlighting current challenges and potential solutions for the advancement of watermarking technology in the era of deep learning.

2. Fundamentals of Digital Watermarking

2.1. Watermarking Workflow

The watermarking process consists of two fundamental stages: watermark embedding and watermark extraction, as illustrated in Figure 2.

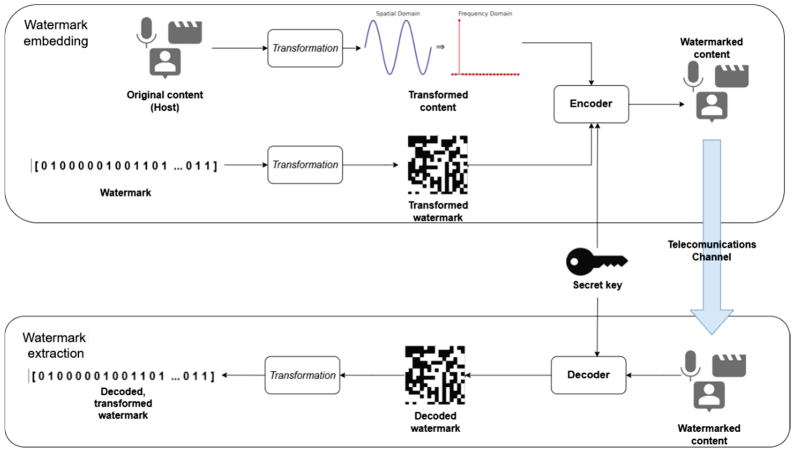


Figure 2. The watermark embedding and extraction scheme, with a secret key controlling the embedding and detection processes.

The watermark embedding process begins by transforming the original content (host) into a selected domain (e.g., the frequency domain), where the watermark is inserted. Additional transformations may be applied to further enhance the method's effectiveness. Optionally, the watermark itself can be transformed depending on the method's requirements, content type, and intended application. Digital signal processing (DSP) algorithms, by operating on signal parameters such as the phase angle, can embed additional information into the useful signal [16,17], which is particularly valuable in the watermarking of digital objects. During embedding, a chosen algorithm or architecture (encoder) introduces the watermark into the host through minor modifications to its content. The watermarked content is then transformed back into its original domain and transmitted through a telecommunication channel. During transmission and subsequent processing, it may be exposed to intentional or unintentional attacks that could remove or distort the watermark. The extraction process follows a similar approach. A decoding algorithm (decoder), compatible with the embedding method, processes the content in order to retrieve the embedded data. If the watermark underwent transformations or encryption during the embedding stage, inverse transformations are applied to restore it to its original form.

https://doi.org/10.3390/s26020444

The described mechanism forms the foundation for a wide range of practical applications where effective and durable protection of digital content is required. Digital watermarking supports a broad spectrum of use cases depending on the target medium and protection goals. In commercial and legal contexts, it enables copyright protection and ownership verification. In the medical domain, it helps ensure the authenticity and traceability of diagnostic images. In military and IoT systems, watermarking contributes to data integrity and access control. The use cases illustrated in Figure 3 encompass both traditional applications (e.g., document security and media monitoring) and evolving needs in intelligent systems (e.g., teleconferencing and remote education environments).

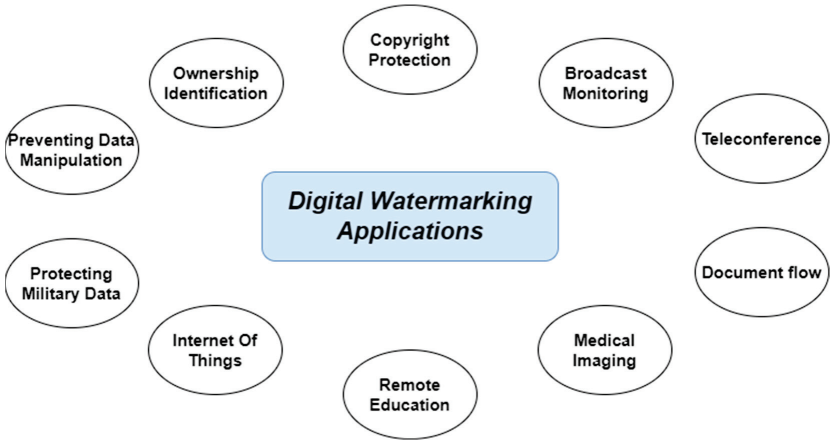


Figure 3. Representative applications of digital watermarking.

2.2. Watermarking Taxonomy

Digital watermarking can be classified according to various criteria. Figure 4 presents a synthesized taxonomy derived from multiple classification schemes reported in the literature, complemented with additional author-defined elements to reflect recent developments and practical perspectives.

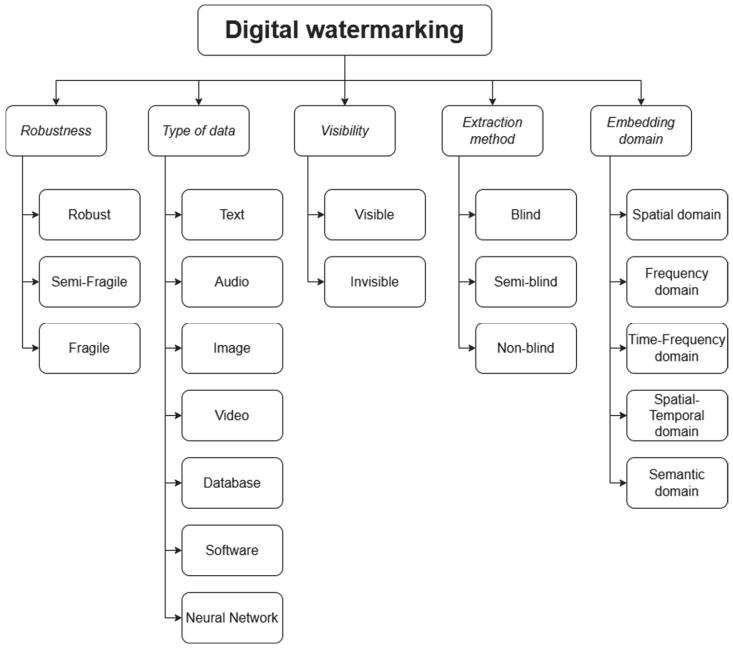


Figure 4. Synthesized taxonomy of digital watermarking methods based on prior literature.

https://doi.org/10.3390/s26020444

Based on their resistance to attacks, digital watermarking methods can be classified into robust, semi-fragile, and fragile categories. In most practical applications, robust methods are used, as they are designed to ensure that the watermark survives typical media processing operations and intentional attacks aimed at damaging or removing the watermark [18]. Semi-fragile methods are usually resistant to processing operations but not to more intensive modifications or deliberate attacks. They are intended for systems where minimal interference with quality is crucial, and intensive data processing is not anticipated. Fragile systems are intentionally designed so that the data is destroyed or damaged with any modification of the carrier. They are used for detecting manipulations and forgeries in the case of highly sensitive data.

The watermark can be embedded in various carriers, the most common being audio data [19–21], text [22–24], images [25–27], and video [28,29]. Due to technological advancements and the need to protect creators in emerging fields, more advanced forms of watermarking have appeared, such as software watermarking [30,31], database watermarking [32,33], neural network model watermarking [34,35], as well as watermarking techniques applied to digital 3D objects [36]. Methods used in these areas include code obfuscation [37], modification of neural network weights [38], and the use of specially prepared trigger sets, which do not affect the model's functionality but enable the activation of the watermark under specific inputs [39].

Based on visibility, watermarks are classified as visible and invisible. Visible watermarks, such as a company logo, are placed in a way that is noticeable but does not hinder content consumption. In contrast, invisible watermarks are designed to be completely imperceptible to the viewer.

During the watermark extraction procedure, typically only the carrier with the embedded watermark is required; such methods are referred to as blind watermarking. If additional information is necessary for extraction, the methods are classified as semi-blind and non-blind. In semi-blind methods, reference data or a key used during watermark embedding is usually required, such as an attention mask. In non-blind methods, the use of the original data carrier is essential, and the extraction process involves comparing the watermarked data with the original content.

In practical watermarking systems, the use of secret keys is very popular because they play a key role in ensuring security, especially when it is assumed that embedding and extraction algorithms are widely known. The watermark key controls both the embedding and detection processes and is the primary mechanism for preventing unauthorized insertion or extraction of watermarks. Depending on the system design, the same key may be used for both embedding and detection, or separate keys may be used. This key-based security assumption is particularly important in semi-blind watermarking schemes, which dominate real-world applications.

There are two primary domains for watermark embedding: the spatial domain, where the watermark is inserted directly into image pixels or audio samples, and the frequency domain, where the watermark is embedded into the coefficients of a selected transform, allowing, among other benefits, higher resistance to lossy compression. To improve efficiency, hybrid domains are also used, typically combining the advantages of both the spatial and frequency domains [40]. Among the hybrid approaches, the time-frequency domain enables better performance for dynamic signals such as audio or video, while the time-spatial domain, mainly used for video, allows for embedding the watermark both statically and in temporal changes between frames. The semantic domain can be considered a variant of the spatial domain, where watermarks are embedded in areas with specific semantic significance to minimally impact perception.

https://doi.org/10.3390/s26020444

The remainder of this article focuses on image watermarking methods that ensure robustness and invisibility.

2.3. Watermarking Paradigms and Metrics

In the commercial watermarking applications, it is essential for the technology to meet three fundamental paradigms that determine system's effectiveness and quality: robustness, transparency, and bit capacity. These characteristics are interdependent, making it difficult to improve one criterion without affecting the others.

2.3.1. Transparency

Transparency, or the invisibility of the watermark to the human visual system, is a key feature that often determines whether a given technology will be implemented. It is assessed using both objective metrics, which measure differences between the original carrier and the carrier with the embedded watermark, and subjective metrics, which evaluate content quality based on user perception.

Objective metrics

The most commonly used objective transparency metrics applied to images are listed below:

- MSE (Mean Squared Error)—the average squared error between the pixel values of the original image and the watermarked image.

$$\mathrm{MSE} = \frac{1}{\mathrm{mn}} \sum_{0}^{\mathrm{m}-1} \sum_{0}^{\mathrm{n}-1} \|\mathrm{f(i,j)} - \mathrm{g(i,j)}\|^2, \tag{1}$$

f—the matrix data of the original image,
g—the matrix data of the watermarked image,
m—the number of pixel rows in the images and i represents the index of that row,
n—the number of pixel columns in the image and j represents the index of that column.

- PSNR (Peak Signal-to-Noise Ratio)—measures the ratio of the original image signal to the noise introduced by the watermarking process.

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\mathrm{MAX_f}}{\sqrt{\mathrm{MSE}}}\right), \tag{2}$$

MAX—the maximum possible value of a pixel (e.g., 255 for an 8-bit image).

- SSIM (Structural Similarity Index) [41]—measures the structural similarity between two images by analyzing contrast, brightness, and texture. The values range from 0 to 1, with values closer to 1 indicating higher similarity.

$$\mathrm{SSIM(x,y)} = \frac{\left(2\mu_x\mu_y + \mathrm{C_1}\right)\left(2\sigma_{xy} + \mathrm{C_2}\right)}{\left(\mu_x^2 + \mu_y^2 + \mathrm{C_1}\right)\left(\sigma_x^2 + \sigma_y^2 + \mathrm{C_2}\right)}, \tag{3}$$

μ_x, μ_y–mean luminance values for X and Y images,
σ_x σ_y—standard deviation values for X and Y images,
σ_{xy}—covariance value between X and Y images,
C_1 C_2—stabilizing constants to prevent division by zero, where

$$\mathrm{C_1} = (\mathrm{K_1 L})^2, \tag{4}$$

$$\mathrm{C_2} = (\mathrm{K_2 L})^2, \tag{5}$$

https://doi.org/10.3390/s26020444

L—dynamic range of the pixel values (255 for 8-bit grayscale images),
K_1, K_2—small constant, $K_1 \ll 1$ and $K_2 \ll 1$.

- MS-SSIM (Multiscale Structural Similarity Index) [42]—an extension of SSIM that considers multiple spatial scales, calculated through a multi-stage downsampling process.

$$\mathrm{MS_SSIM}(x, y) = \prod_{j=1}^{M} \left[\mathrm{SSIM}_j(x, y)\right]^{\alpha_j}, \tag{6}$$

M—number of scale levels,
α_j—weights (usually $\alpha_j = 1/M$),
$SSIM_j(x, y)$—SSIM value at the j level.

- VIF (Visual Information Fidelity)—measures the amount of visual information transferred from the original image to the watermarked image using the HVS (Human Visual System) model.

$$I_0 = \sum \log_2\left(1 + \frac{\sigma^2_{x_k}}{\sigma^2_{n_k}}\right), \tag{7}$$

I_0—the amount of information in the original channel,
$\sigma^2_{x_k}$—the variance of signal at the k level,
$\sigma^2_{n_k}$—the variance of noise at the k level.

$$I_w = \sum \log_2\left(1 + \frac{\sigma^2_{x_k}}{\sigma^2_{n_k} + \sigma^2_{e_k}}\right), \tag{8}$$

I_w—the amount of information in the modified channel,
$\sigma^2_{e_k}$—the variance of error resulting from the difference between the original and the modified image.

$$\mathrm{VIF} = \frac{I_w}{I_0}, \tag{9}$$

- FSIM (Feature Similarity Index)—compares key visual features from the perspective of human perception.

$$\mathrm{FSIM}(x, y) = \frac{\sum_{i \in \Omega} PC_m(i) \cdot S_L(i) \cdot S_P(i)}{\sum_{i \in \Omega} PC_m(i)}, \tag{10}$$

$PC_m(i)$ —phase congruence at a point i,
$S_L(i)$—luminance similarity function,
$S_P(i)$—phase congruence similarity function.

Subjective Metrics

Subjective metrics are used to obtain direct user assessments, where individuals evaluate the quality of an image or video based on their own perception. Testers determine the extent to which content modifications are noticeable and how they affect the overall experience. The conditions and methodology for conducting these tests are thoroughly described in the International Telecommunication Union recommendations, specifically for television images [43] and video materials in lower bandwidth applications such as videoconferencing [44]. However, the methodology can also be successfully applied to static images:

https://doi.org/10.3390/s26020444

- DSIS (Double Stimulus Impairment Scale)—the method involves comparing two versions of the same material: the original reference version and the processed version. Users view both versions sequentially and then assess the degree of quality degradation in the processed version relative to the original. Ratings are collected on a quality scale from 1 to 5, where 5 indicates that the differences between the original and the modified content are imperceptible, and 1 signifies that the content quality has significantly deteriorated and is unacceptable.
- DSCQS (Double Stimulus Continuous Quality Scale)—users are presented with both the original and the modified versions of the content, but they are not explicitly informed which is the original or the modified version. Similar to DSIS, users evaluate the content quality; however, the lack of clarity regarding which material has been altered provides a more objective assessment from the perspective of human perception.
- Paired Comparison Test—users are shown two versions of the content: the original and the modified, without indicating which one has been altered. Participants evaluate which version they believe has higher quality or whether they can notice any differences.

AI-Based Perceptual Metrics

With the development of deep learning techniques, modern perceptual metrics utilizing neural networks have emerged, offering a better reflection of human image perception compared to classical similarity metrics [45]. Selected deep learning-based metrics used to evaluate the transparency of watermarking systems include:

- LPIPS (Learned Perceptual Image Patch Similarity) [46]—this method employs convolutional networks trained on large datasets to measure perceptual differences between images. Metric values close to 0 indicate smaller differences and higher transparency of the method.
- DISTS (Deep Image Structure and Texture Similarity) [47]—a metric that combines texture and structure analysis using deep features from neural networks, providing improved assessment of images with complex details.
- PieAPP (Perceptual Image-Error Assessment through Pairwise Preference) [48]—a trained model that predicts image quality based on user preferences by evaluating pairs of images.

2.3.2. Robustness

Watermark robustness refers to the system's ability to retain the watermark within the carrier even after processing operations such as compression, filtering, scaling, or intentional attacks aimed at removing the watermark. The assessment of robustness involves comparing the quality of the original watermark embedded in the carrier with the watermark extracted after processing operations. Among the robustness evaluation metrics, the following are distinguished:

- BER (Bit Error Rate)—measures the percentage of bits that have been incorrectly extracted from the embedded watermark compared to the original watermark.

$$\mathrm{BER} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left| \mathrm{w_i} - \hat{\mathrm{w_i}} \right|, \tag{11}$$

w_i—value of the i-th bit in the original watermark,
$\hat{w_i}$—value of the i-th bit in the extracted watermark,
N—total number of bits.

https://doi.org/10.3390/s26020444

- NC (Normalized Correlation)—a metric that measures the similarity between the original and the extracted watermark. A value close to 1 indicates high resistance to attacks.

$$NC = \frac{\sum_{i=1}^{N} w_i \cdot \hat{w_i}}{\sqrt{\sum_{i=1}^{N} w_i^2} \cdot \sqrt{\sum_{i=1}^{N} \hat{w_i}^2}}, \quad (12)$$

- ASR (Attack Success Rate)—it represents the percentage of successful attempts to weaken or remove the watermark as a result of adversarial attacks [49].
- RGA (Robustness Against Generative Attacks)—a metric evaluating the system's resistance to attacks utilizing generative models. It analyzes the extent to which the watermark remains intact after being processed by these models [50].
- APT (Adversarial Perturbation Tolerance)—defines the minimum level of perturbations introduced by adversarial attacks necessary to successfully remove or distort the watermark. A higher APT value indicates greater system robustness [51].

2.3.3. Capacity

Watermark capacity refers to the amount of information that can be embedded in the carrier while maintaining adequate transparency and robustness. Below are the most commonly used capacity metrics:

- Payload Capacity—measures the number of watermark bits relative to the given carrier.

$$P = \frac{N_{bit}}{N_{host}}, \quad (13)$$

N_{bit}—number of bits embedded as watermark,
N_{host}—number of host units (pixels for images, seconds for video or audio).

- Embedding Capacity Efficiency—measures the efficiency with which the system utilizes the carrier's space for embedding the watermark, taking into account the impact on quality and robustness.

$$ECE = \frac{N_{bit}}{N_{max}} \cdot 100\%, \quad (14)$$

N_{max}—maximum capacity of the host, i.e., how many bits can be embedded before the quality of the content is noticeably degraded.

2.4. Attacks on Watermarking Systems

In practical applications, watermarking systems are exposed to various processing operations and deliberate attempts to remove or distort the embedded watermark. The previously described metrics measure how well a watermarking system withstands attempts to destroy or alter the watermark. However, different types of attacks can cause diverse distortions to the carrier and the embedded watermark, making the analysis of their effects crucial for a comprehensive system evaluation. Below is an overview of the main types of attacks along with their impact on carrier quality and watermark integrity, as summarized in Table 1. The attacks presented in the table are divided into three main categories: untargeted attacks including standard multimedia processing operations, targeted attacks; and deep learning attacks. The last category is distinguished as a separate group due to its dynamic development, high effectiveness, and the difficulty in counteracting such attacks.

https://doi.org/10.3390/s26020444

Table 1. Types of attacks on watermarking systems.

Group of Attacks	Type of Attacks	Description	Examples of Operations	Effect on the Watermark
Untargeted attacks	-	Attacks resulting from routine processing of the media, with no intention of removing the watermark, but which may affect its integrity.	Lossy compression (JPEG, MPEG), scaling, filtering.	Partial loss or distortion of the watermark.
Targeted attacks	General	Intentional manipulation of the media to remove, distort, or weaken the watermark.	Rotate, scale, change resolution.	Total or partial loss of the watermark.
Targeted attacks	Statistical attacks	Modifications using statistical analysis of the media to identify and remove the watermark.	Histogram attack, frequency distribution analysis, autocorrelation attack.	Removal of the watermark without significant changes in the perception of the media.
Targeted attacks	Sensitivity attacks	Minimal modifications to the media that do not affect the visual quality, but destroy the watermark.	Bit depth reduction, delicate pixel changes.	Distortion or complete loss of the watermark without visible changes in the media.
Targeted attacks	Destructive compression	Aggressive compression to remove the watermark by extremely reducing the media data.	High-loss JPEG compression.	Significant loss of data, complete destruction of the watermark, degradation of media quality.
Targeted attacks	Geometry attacks	Manipulations in the spatial structure of the media that distort the position of the watermark.	Rotation, translation, change of proportions.	Disturbance of the watermark position, loss of synchronization.
Deep learning-based attacks	Generative attacks	Use of generative models to regenerate media content and remove embedded watermark.	Image inpainting, deepfake generation, AI-based restoration.	Complete removal of the watermark without perceptual changes.
Deep learning-based attacks	Adversarial attacks	Modifications generated by neural networks to fool detection systems and weaken watermark extraction.	Adversarial noise, gradient-based attacks (FGSM, PGD).	Degradation or undetectability of the watermark.
Deep learning-based attacks	Neural network removal	Use of DL models trained to detect and remove watermarks.	CNN-based watermark removal, encoder–decoder architectures.	High probability of the watermark elimination with minimal distortion.
Deep learning-based attacks	Content replacement	Media content is regenerated using deep learning models to overwrite or bypass the watermark layer.	GAN-based texture replacement, style transfer techniques.	Loss or severe weakening of the watermark.
Deep learning-based attacks	Latent-space attacks	Attacks that exploit generative models (diffusion or VAE-based) to regenerate content in the latent space, effectively removing embedded watermarks.	Stable Diffusion regeneration, DERO [52], VAE sampling attacks [53].	Total removal of the watermark, especially in latent-domain watermarking schemes.

Deep learning-based attacks, especially those exploiting latent space regeneration (e.g., via diffusion models or VAEs), tend to be significantly more effective than traditional signal-domain manipulations. The high effectiveness of these attacks stems from the fundamental operating principles of modern generative models. Diffusion models and variational autoencoders learn compact latent representations that capture the semantic structure of the image while discarding high-frequency or weakly correlated signal components. As a result, watermarks embedded in the pixel or frequency domain are often treated as noise during the generative reconstruction process and are not preserved when the image is regenerated from the latent space, yielding visually faithful content devoid of the original watermark.

Unlike traditional signal-domain attacks, which typically apply local distortions or global transformations to the original signal, latent-space regeneration reconstructs the image from learned data distributions. Consequently, synchronization-based and redundancy-based watermarking schemes-effective against compression or geometric attacks-become

https://doi.org/10.3390/s26020444

ineffective when the original content is replaced by a newly synthesized instance that preserves perceptual quality but not the embedded watermark.

This highlights the growing need for DL-robust watermarking strategies that are explicitly designed to counter generative and latent-space attacks. Recent research approaches include embedding watermarks directly into latent representations, integrating watermark-preservation constraints into the training objectives of diffusion or GAN-based generators, and developing joint generation–watermarking frameworks in which watermark survival becomes an inherent property of the synthesis process. Such strategies aim to shift watermarking from a post-processing operation to an integral component of content generation, thereby improving resistance to regeneration-based attacks.

3. Traditional Image Watermarking Methods

Traditional watermarking methods are primarily based on modifying the carrier data in the spatial and the frequency domain, as well as on hybrid-domain approaches that aim to combine the advantages of both, as extensively discussed in the classical watermarking literature [54].

3.1. Spatial Domain

Spatial domain methods are among the simplest and oldest techniques in digital watermarking. They rely on direct modification of pixel values in images or video frames without prior transformation into another domain. A well-known example is the LSB (Least Significant Bit) technique proposed by Turner [55], in which the least significant bits of pixels are modified to hide information. These methods are characterized by high capacity and low complexity but relatively low resistance to attacks. One of the first digital watermarking approaches utilizing this technique was proposed in 1994 [56]. Over the years, the method has undergone numerous modifications. In [57], the authors proposed using the third and fourth LSBs to improve data embedding. Subsequently, in [58], a combination of the LSB method with binary value inversion of the watermark was introduced, enhancing the method's transparency. Arya and Saharan [59] increased both the robustness and transparency of the LSB method by generating the watermark image from the host image and securing it with a key derived from the same source. In [60], the authors modified the traditional approach by introducing a hashing mechanism for the watermark before embedding, which improved resistance to attacks. Another approach combines the LSB method with edge detection techniques [61] to identify suitable regions for watermark embedding and additionally encrypt the watermark to increase the method's security.

The LSB method can also be applied to video signals when watermarking is performed on a frame-by-frame without considering temporal dependencies. In [62], the authors used LSB to embed the watermark into selected video frames and employed an FPGA architecture to accelerate the embedding and extraction processes in real-time. Similarly, in [63], the watermark was embedded into specific video frames using the LSB algorithm, with the frame selection based on histogram analysis.

Among other methods based on pixel value modification, the following can be distinguished: Pixel Value Differencing (PVD) [64,65], the Patchwork algorithm [66–68], Singular Value Decomposition (SVD) [69,70], and the Arnold transformation [71,72].

3.2. Frequency Domain

In frequency-domain watermark embedding methods, the content is transformed using a selected transform, after which the watermark is embedded into the transform coefficients. Once embedded, the watermarked content is transformed back into the original domain.

https://doi.org/10.3390/s26020444

This approach generally exhibits higher resistance to content processing operations, such as compression, scaling, and deliberate watermark attacks. In the case of images, watermark embedding most commonly utilizes the Discrete Cosine Transform (DCT) [73–78], the Discrete Fourier Transform (DFT) [79–82], and the Discrete Wavelet Transform (DWT) [83–86]. Transform-based methods offer numerous advantages over spatial domain watermarking approaches. DCT-based solutions are known for their high resistance to compression. DWT, owing to its multilevel analysis capabilities, increases resistance to basic processing operations, while DFT enables watermark embedding in frequency ranges that are less susceptible to modifications, making the watermark more robust against geometric transformations. Combinations of different transforms enable the integration of their individual strengths, resulting in solutions with greater robustness and higher transparency, such as DCT–DWT [87–90], DFT–DCT [91–93], DFT–DWT [94–96], and DFT–DCT–DWT [97].

Similar to the spatial domain approaches, it is possible to apply one-dimensional or two-dimensional transforms, including video signals, provided that watermarking is performed on individual frames without considering temporal context. This approach has been proposed in the following publications for the transforms: DCT [98–102], DWT [103–106], 1D-DFT [107], 2D-DFT [108,109], as well as their combinations, which aim to improve the method's efficiency [110–113].

3.3. Hybrid Domains

Typical hybrid domains, such as the time-frequency and the time-spatial domain, are mainly applied in video signal watermarking. In the case of images, the semantic domain can be used, which is classified as hybrid since it combines elements from both the spatial and frequency domains. However, its key feature is content-level analysis at the semantic level. The watermark is embedded in selected areas of the image or video that are significant from a content perspective. This approach enables the watermark to be hidden in regions that are perceptually or semantically important to the user, making the watermarking process both more resistant to manipulation attempts and less noticeable. Semantic domain methods rely on various image processing techniques, such as segmentation, edge detection, and object detection [114].

3.4. Summary

In summary, the classification of traditional watermarking methods for images and video frames is presented in Figure 5.

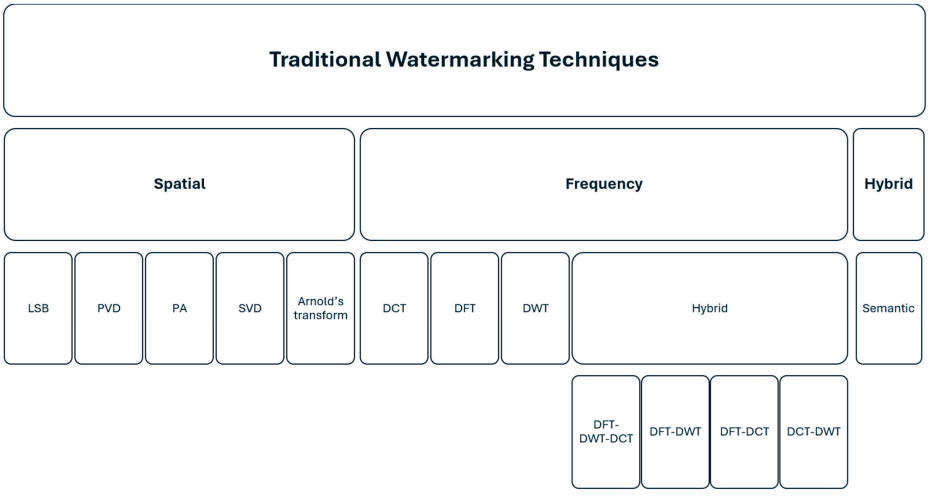


Figure 5. Taxonomy of traditional watermarking algorithms for images and video frames. Spatial-domain methods include LSB [56–63], PVD [64,65], PA [66–68], SVD [69,70], and Arnold's transform [71,72].

https://doi.org/10.3390/s26020444

Frequency-domain methods include DCT [73–78,98–102], DFT [79–82,107–109], and DWT [83–86,103–106]. Hybrid approaches combining DFT–DWT–DCT [97,113], DFT–DWT [94–96], DFT–DCT [91–93], and DCT–DWT [87–92,110–112] are also shown. Semantic watermarking methods are presented in [114].

4. Deep Learning-Based Watermarking

In recent years, the rapid development of deep learning-based technologies has revolutionized many areas of science, including digital watermarking. Unlike traditional methods, which rely on manually designed features, deep learning algorithms learn optimal data representations, enabling more efficient, robust, and invisible watermark embedding.

Deep learning is a subset of machine learning that utilizes multi-layer neural networks (deep neural networks) [115,116]. Due to their architecture and ability to automatically recognize and learn patterns, DL algorithms can learn to be robust to various types of targeted and untargeted attacks by appropriately defining loss functions. Additionally, neural networks are highly efficient in solving problems that require scalability. This is due to the hierarchical structure of feature learning, the ability to process in parallel, and support for multi-dimensional data [117], making neural networks very effective even in watermarking high-resolution images and videos, such as 4K resolution.

4.1. Deep Learning Architectures Used in Image Watermarking

The fundamental architectures used in image watermarking are convolutional neural networks. These are a type of deep neural networks specifically designed for analyzing data with a matrix-like structure, such as images or video frames, which has ledthem to dominate the processing of such data for years [118]. The network consists of convolutional blocks, which include convolutional layers, activation layers, and pooling layers. The key component are the convolutional layers, which consist of filters whose primary function is to extract features from images, enabling edge detection, texture identification, and the recognition of more complex patterns. In watermarking, CNNs are utilized in watermark encoder and decoder algorithms. The convolutional filters learn to modify input images in such a way that the watermark is embedded, ensuring that the watermark is both transparent and resilient to attacks. The primary advantage of such CNN-based solutions is their simplicity of implementation and flexibility in adapting to different input data and watermarks. The general block diagram of a convolutional network is shown in Figure 6.

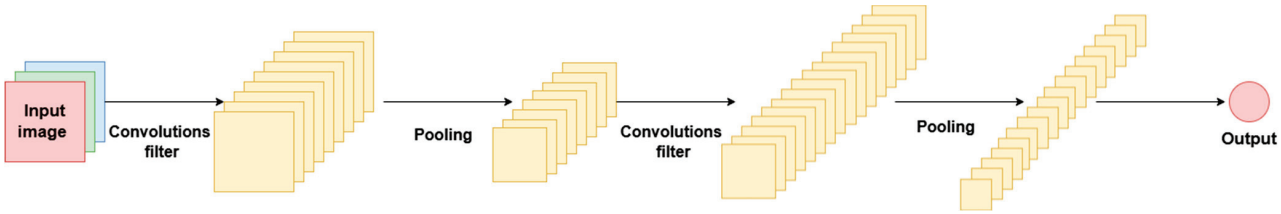


Figure 6. Convolutional Neural Network schema.

Although CNNs form the basis of many watermarking algorithms, to achieve algorithms that meet more stringent requirements in terms of robustness and transparency, it is also necessary to use other architectures. Autoencoders provide an optimized approach to data encoding and decoding data. Their architecture allows data to be transformed into a different (hidden) space, followed by data reconstruction [119]. An autoencoder consists of two parts—Figure 7:

- Encoder, which transforms the input data into a lower-dimensional representation, with the goal of reducing the data size while capturing the most important features of the input,

https://doi.org/10.3390/s26020444

- Decoder, which reconstructs the data based on the representation by applying reverse transformations to those used in the encoder.

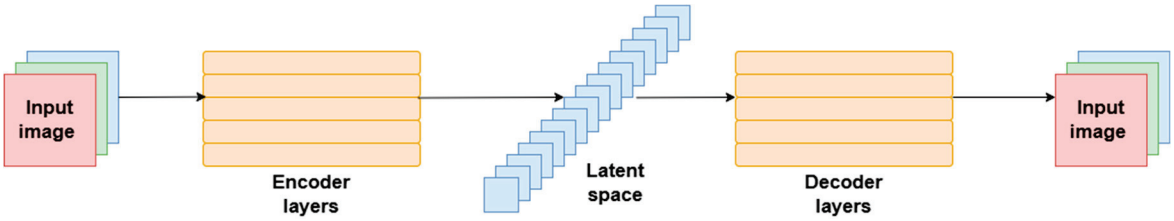


Figure 7. Autoencoder schema.

Autoencoders are trained in an unsupervised manner, minimizing the difference between the input and output data by using an appropriately selected cost function [120–122]. The architecture is a natural choice for watermarking tasks because the encoder–decoder structure mirrors the process of embedding and extracting a watermark.

Another approach for creating more efficient watermarking algorithms is the use of Generative Adversarial Networks. This architecture relies on two competing neural networks [123]—Figure 8:

- Generator, based on provided features or random noise, learns to generate new data,
- Discriminator attempts to distinguish between real data and data generated by the generator.

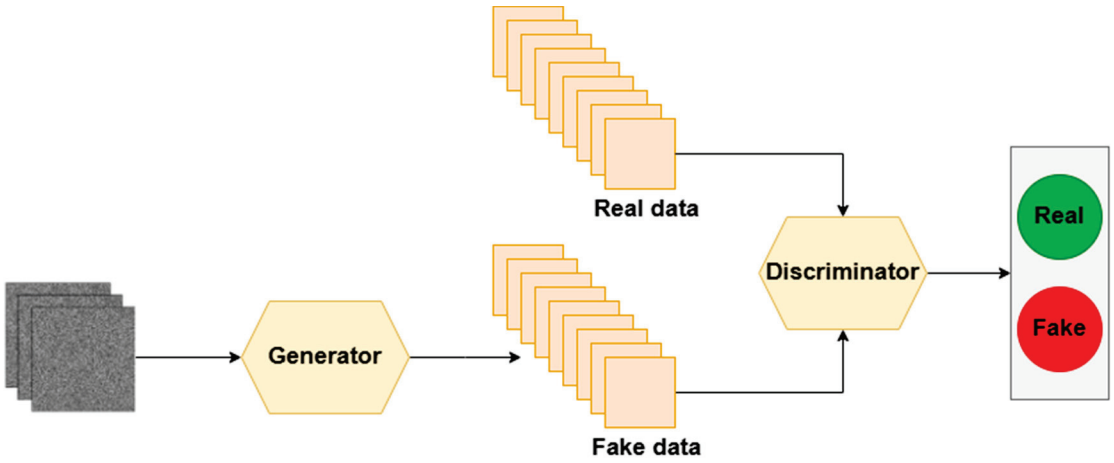


Figure 8. Generative Adversarial Networks schema.

In watermarking applications, an extension of this concept is typically used, namely DCGAN (Deep Convolutional Generative Adversarial Networks)—a GAN architecture built on convolutional networks [124]. The generator learns to embed the watermark into the provided data in an invisible manner, while the discriminator, acting as a critic, evaluates whether the watermark has been properly hidden. Due to their operating principles, GANs enables the development of algorithms characterized by high transparency and are also easily adaptable to different types of data.

The main drawback of GANs is their instability, as the competition between the generator and the discriminator makes them quite difficult to train. A potential solution to these problems lies in the use of diffusion models. These are generative structures that, through the use of a reverse diffusion process, learn to generate or reconstruct data from random noise [125,126]. The input data is iteratively noised in a controlled manner until it becomes completely random noise. Then, the model learns the reverse process—recovering the original data from the noisy

https://doi.org/10.3390/s26020444

data. The conceptual diagram of the model is shown in Figure 9. Diffusion models are not yet widely adopted in watermarking solutions, but due to their noise removal mechanism, they can be highly effective in extracting watermarks even after various types of attacks.

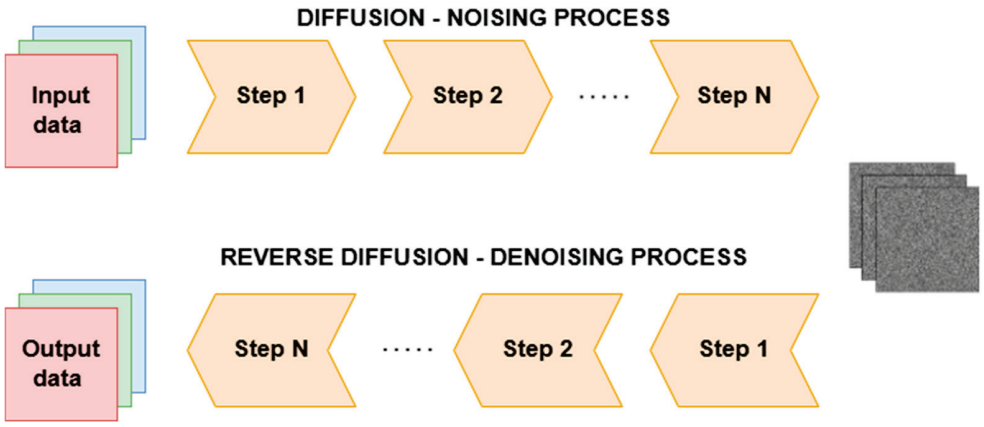


Figure 9. Diffusion model schema.

Currently, Transformer models are increasingly being used, surpassing the efficiency of their predecessors. A Transformer architecture consists of two components: an encoder with an attention mechanism that transforms the data into an internal representation, and a decoder with an attention mechanism that, based on the input representation and the target sequence, predicts the next sequence elements with a given probability [127]—Figure 10.

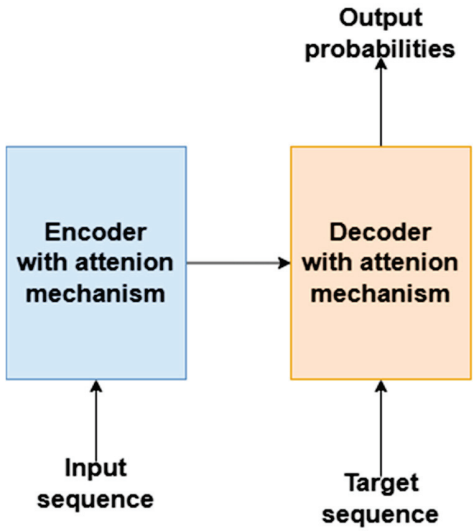


Figure 10. Transformer schema.

The central element of Transformers is the attention mechanism, which enables the capture of dependencies between components of the input sequence, regardless of their relative distance. This mechanism, known as self-attention, analyzes how each element of the sequence is related to the others. In image watermarking applications, a specialized version called spatial attention [128] is used, which focuses on spatial relationships in the input data, such as between pixels or pixel blocks in images. The spatial attention mechanism helps identify key areas for analysis, such as the optimal location for watermark embedding.

For image processing applications, a dedicated variant of the Transformer, the Vision Transformer (ViT), was developed. Unlike traditional convolutional neural networks (CNNs), ViTs divide the image into smaller fragments (called patches), which are then

https://doi.org/10.3390/s26020444

transformed into vectors and processed through the self-attention mechanism. This approach allows the model to capture global dependencies within the image. While Vision Transformers [129] can achieve higher performance compared to CNNs, they require a larger amount of training data to be effective.

4.2. Overview of Deep Learning-Based Image Watermarking Algorithms

Given the diversity of visual data, varying resolutions, and applications requirements, different solutions are employed, each tailored to specific problems. These solutions have been described and classified in literature reviews [130–132].

The first attempts to apply deep learning in image watermarking emerged around 2017. One of the initial approaches was based on a CNN that operates similarly to an autoencoder [133]. Two independent CNNs generate two image sets, created as a codebook, which are then permuted using cryptographic keys. For each bit of the watermark, an appropriate pair of codebook codes is selected and embedded into the image. This method enables the embedding of a 64 $\times$ 64-pixel watermark into a 128 $\times$ 128-pixel image, offering resistance to common image processing attacks and JPEG compression while ensuring a high level of security due to the use of cryptographic keys. In [134], the authors proposed a blind watermarking method based on CNNs using an end-to-end approach in which embedding and extraction processes are optimized together within a single architecture. The neural network embeds a 1-bit watermark in each sub-block of the image with dimensions 8 $\times$ 8 pixels. Subsequently, selected geometric attacks and signal processing operations are simulated to enhance the algorithm's robustness.

In subsequent work, architectures based on the use of two or three main modules—a watermark encoder, decoder, and, optionally, a module simulating attacks on the watermark—became dominant. In [135], the authors proposed an architecture that enables embedding a watermark in the form of an audio file into an image. Two neural networks were utilized: WM Network and Similarity Network. The WM Network consists of the Encoder and Decoder modules, which map the watermark using LSTM, and Embedder and Extractor, which are based on convolutional layers. Similarity Network compares the original watermark with the extracted one, allowing the evaluation of the method's effectiveness. A similar approach was adopted in [136]. This is a blind algorithm based on CNNs, designed to extract the watermark from images captured using mobile phones. The architecture includes components responsible for mapping and demapping the watermark, as well as its embedding and extraction, all composed of convolutional layers. A key element of the algorithm is the Invariance Layer, which affects the algorithm's resistance to attacks. The module is based on fully connected layers and allows for dispersing information across different parts of the image.

In [137], the authors also used a CNN-based encoder and decoder, but during the preprocessing of the host image, they applied the Wavelet Transform and an additional convolutional network responsible for watermark preprocessing. Moreover, the authors implemented an attack simulator to enhance the algorithm's robustness against basic image processing operations. A similar architecture, based on an encoder and decoder composed of convolutional layers along with an attack simulator, was implemented in [138]. The authors employed DWT before embedding the watermark and IDWT (Inverse Discrete Wavelet Transform) after embedding to restore the image to its original domain. The watermark is in the form of a binary image with dimensions of 32 $\times$ 32 pixels, eliminating the need for an additional watermark processing block.

Lu et al. [139] also used the Wavelet Transform but incorporated it within the embedding algorithms. As the encoder, they employed the U-Net autoencoder architecture [140],

 https://doi.org/10.3390/s26020444

in which the central convolutional and deconvolutional blocks were replaced with DWT and IDWT blocks. In the decoder, convolutional layers and DWT blocks were implemented. Between the encoder and decoder blocks, an attack simulation block inspired by StegaStamp was introduced. StegaStamp [141] is a solution that enables the embedding of hyperlinks into images. The encoder uses a convolutional network, also similar to U-Net. A key component of the system is the distortion simulation block, which, unlike most watermarking techniques, enhances the algorithm's resistance to distortions introduced during the printing of images. This feature allows the method to be used in real-world physical scenarios, such as printed photographs or billboards.

In order to increase the transparency of the algorithm, the authors in [142] utilized an autoencoder architecture in both the watermark encoder and decoder. In the encoding algorithm, the watermark embedding occurs in the latent space, after which the image is scaled back to its original resolution. In the decoder, a denoising autoencoder is first used to reduce the effects of noise and other distortions (if present). Subsequently, two encoders are employed to extract the watermark based on both the original image and the image with the embedded watermark. The network training is conducted in two stages.

In subsequent approaches, the encoder–decoder architecture was enhanced with an attention mechanism. Dasgupta and Zhong [143] proposed a solution utilizing the multi-head cross-attention mechanism (MHA) [127], which enables the model to learn mutual dependencies between two different data sequences, in this case, between the watermark and the host image. Additionally, the authors employed representation learning in an invariant domain using a triplet loss function [144], which optimizes the distances between images containing the same watermark content (anchor and positive) while maximizing the differences between them and a negative image (any other image). This approach improves the robustness and the model's ability to learn the watermarking pattern.

The second group of algorithms consists of GAN-based approaches, where a discriminator module is added to the encoder and decoder blocks. The watermark encoder functions as a data generator, while the discriminator acts as a critic, assessing the visibility of the watermark. The first such solution was the HiDDeN architecture [145] proposed by Zhu et al. It is an end-to-end solution based on three CNNs. The encoder embeds a bit sequence into the host image; the watermarked image is then processed through a distortion layer, after which the decoder extracts the watermark. The discriminator verifies whether the watermarked image is sufficiently similar to the original image, allowing the method to achieve high transparency.

Wen and Aydore proposed an improvement to this approach by developing the ROMark algorithm [146]. The authors utilized the HiDDeN architecture, introducing min-max optimization, which involves optimizing losses under the worst-case scenario. This is achieved through the implementation of a dynamic noise layer that iteratively generates the most challenging possible distortions. Additionally, the range of applied distortions was expanded, and gradient propagation was improved to facilitate more efficient network training. The HiDDeN algorithm was also used in [147], where the authors extended the distortion module by adding a rotation layer and an additive noise layer. They also modified the loss function, which enabled a better trade-off between robustness and transparency. In [148], the authors similarly focused on optimizing the learning process proposed in [145]. They introduced a two-stage training process, training the encoder and decoder for the base model, and training separate decoders for different types of distortions in the second stage to better enhance the watermark's resistance to various types of attacks.

A completely different GAN-based model was proposed in [149]. The authors utilized the Inverse Gradient Attention (IGA) mechanism, which dynamically identifies image pixels most resistant to distortions and assigns them higher weights during the message-hiding

https://doi.org/10.3390/s26020444

process. Additionally, the encoder module enables the compression of binary messages into real numbers, allowing the embedding of a larger number of bits (256 bits) without affecting image transparency. A similar approach, employing a different algorithm, was presented by Hao et al. [150]. The authors applied attention mechanisms in both the encoder (generator) and decoder to identify areas most resistant to disturbances and focus on them during watermark embedding. The implemented discriminator not only serves as a critic but also evaluates the visual quality of images and supports robustness optimization. A dynamic disturbance layer was also incorporated to simulate multiple attacks simultaneously. Attention mechanisms were further utilized in [151]. The ARWGAN model employs an attention mechanism in the encoder to generate an attention mask, enabling the placement of the watermark in the most optimal image regions. Additionally, a Feature Fusion Module was used to extract image features and leverage them to enhance robustness. The authors also implemented a Noise Subnetwork to simulate various types of distortions. In [152], the authors applied a GAN-LSTM structure combined with the Adaptive Gannet Optimization algorithm, which, similar to previous solutions, enables the selection of optimal locations for watermark embedding. The watermark undergoes preprocessing using DWT and Schur decomposition and is subsequently chaotically encrypted. Incorporating the LSTM architecture into the algorithm allows for better management of the watermark extraction process, improving both accuracy and robustness.

The previously described attention-based solutions enabled the dynamic identification of optimal image regions for watermark embedding. An extension of this concept involves the use of more advanced architectures, such as Transformers, which bring new capabilities to watermarking-related challenges. In [153], the authors implemented Transformers for both text processing and visual Transformers for image feature extraction as part of the proposed text embedding algorithm. The text is encoded into a vector representation with dimensions 16×64 using a Transformer, while the encoder and decoder are based on the ViT architecture. To improve algorithm robustness, noise was added to the encoded text representation, and common image distortions were applied to the watermarked image. In [154], a ViT was also employed for image authentication, manipulation detection, and image recovery. The source image undergoes preprocessing using several transforms: Discrete Wavelet Transform, Integer Wavelet Transform, Schur decomposition, and Curvelet Transform, the latter enabling the identification of high-entropy areas suitable for watermark embedding. The model generates encoded feature maps to serve as watermarks. Additionally, an authentication key is generated using Singular Value Decomposition.

Liu et al. [155] proposed a two-stage approach for watermark embedding using Transformers. The input images are transformed using DWT, enabling decomposition into frequency coefficients, with the watermark embedded in the low-frequency components to increase robustness. The attention module stores information about both low and high frequencies, enhancing the stability and accuracy of image reconstruction. As part of the two-stage embedding procedure, the encoder and decoder are first trained to embed and extract the watermark, and then a reversible information embedding procedure is introduced based on the first stage (freezing the encoder weights) to increase robustness. In [156], a transformer was combined with a GAN structure through the implementation of a discriminator that supports the transparency of the image with the embedded watermark. In the encoder, self-attention-based preprocessing was used to expand the watermark features and increase embedding efficiency by evenly distributing the watermark information across the image. Additionally, a Feature Enhancement Module was utilized to identify relationships between the image and the watermark using cross-attention, and a Soft Fusion Module, responsible for the final fusion of image and watermark features based

 https://doi.org/10.3390/s26020444

on self-attention and cross-attention. The solution also includes a Noise Layer responsible for simulating disturbances and increasing robustness.

In [157,158], the authors employed the Swin Transformer architecture [159], which was designed for image processing and is characterized by better global information flow and lower computational complexity than ViT. In [157], the Swin Transformer was combined with CNN. The watermark encoder is based on CNN and blocks Squeeze-and-Excitation (SE), which support the extraction of important features. In the decoder, CNN was innovatively used to analyze local features, Swin Transformer, which allows for hierarchical feature processing and improves global feature representation, and the Identity module, which, through residual connections, facilitates feature extraction. Additionally, a Multi-scale Attentional Feature Fusion module (MA-FFM) was implemented to iteratively combine global and local features. In [158], the END (Encoder-Noise-Decoder) structure was used. The watermark encoder is based on the U-Net architecture, supplemented with a Locally-Channel Enhanced Swin Transformer Block, which utilizes window-based self-attention, and a Frequency-Enhanced Transformer Block, which applies an attention mechanism in the frequency domain using the cosine transform. The window-based self-attention mechanism involves dividing the image into smaller areas (windows) and analyzing dependencies only within the window area rather than globally across the entire image, followed by window shifting. The decoder structure is similar to the encoder but uses additional attention mechanisms to precisely recover the watermark.

The next significant step in the field of watermarking was the use of diffusion models, which allow for even greater resistance to disturbances and better control over the structure of embedded information. Initial research on watermarking in the context of diffusion models focused primarily on the detection and marking of images generated by these models [160–162]. In the subsequent phase, diffusion algorithms themselves began to be used for embedding and extracting watermarks. One of the first such approaches is the ZoDiac algorithm [163], described by Zhang et al., which utilizes the Stable Diffusion model [164] to embed watermarks in the latent space of the image. The watermark is embedded into the frequency coefficients of the Fourier transform, making it more resistant to attacks and manipulations, including generative attacks (Stable Diffusion-based removal attacks). Stable Diffusion is a text-to-image model developed for text-to-image generation. ZoDiac uses a pre-trained model with a denoising step of 50 to reconstruct the image in the latent space, which reduces the time and resources required to deploy the system.

The WaterDiff method proposed in [165] is also based on a pre-trained diffusion model. The host image is transformed by two separate encoder modules into a latent feature vector of high-level features and a matrix of low-level features. The watermark is then embedded into the coefficients of the wavelet transform, and the final watermarked image is reconstructed into its original form using the diffusion model. The extraction process proceeds analogously and relies on the use of the pre-trained probabilistic model. A distinctive feature of this approach is its very high watermark capacity of 1 bit per pixel (bpp) and the flexible ability to adjust the frequency subspace in which the watermark is to be hidden.

In [166], the SuperMark algorithm is described, which is also based on the use of a pre-trained diffusion model, but dedicated to super-resolution (SR). The watermark embedding process involves transforming the input image to a lower resolution compatible with the SR model input, embedding the watermark in the latent space through Gaussian Shading, and denoising and reconstructing the image with the embedded watermark back to its original resolution. During watermark extraction, the image is again scaled to a lower resolution and transformed into the latent space using variational autoencoder. The use of

https://doi.org/10.3390/s26020444

the diffusion model allows the reconstruction of the original Gaussian noise, from which the watermark bits are extracted.

4.3. Summary of Deep Learning-Based Image Watermarking Algorithms

The development of image watermarking techniques based on deep learning has followed the evolution of neural network architectures. Figure 11 illustrates a historical timeline of key deep learning architectures (marked in green) alongside their applications in watermarking (marked in orange). The watermarking methods referenced here were described in the previous subsection. The emergence of convolutional networks (1980) and their subsequent popularization (2012) revolutionized image processing. Furthermore, the introduction of deep autoencoders (2006) further expanded the capabilities of neural networks, leading to advanced image encoding techniques such as U-Net (2015).

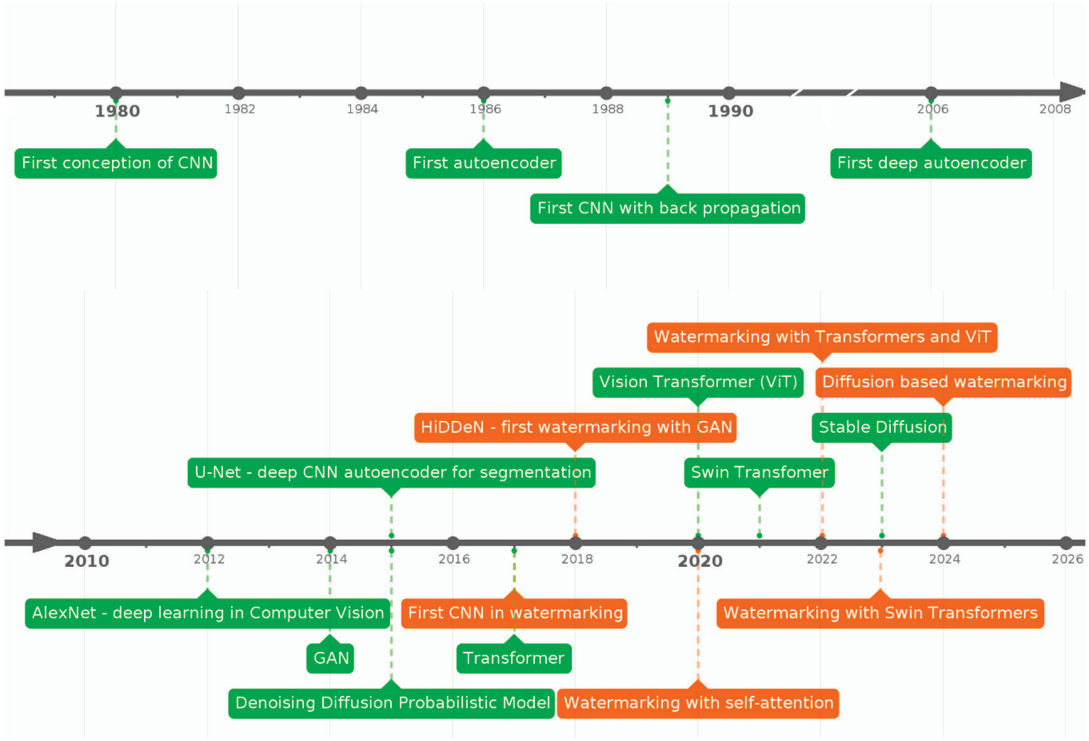


Figure 11. Timeline of development of deep learning-based image watermarking methods.

The first application of CNN in watermarking appeared in 2017, and by 2018, the groundbreaking HiDDeN model was developed—the first deep learning-based watermarking method utilizing GANs. Simultaneously, the introduction of Transformers (2017) and their improvements dedicated to image processing (ViT, 2020; Swin Transformer, 2021) paved the way for watermarking applications leveraging the attention mechanism. In 2022, watermarking methods based on ViT emerged, followed by those utilizing the Swin Transformer in 2023.

The latest trend in image watermarking is the use of diffusion models, which, although introduced in 2015, gained significant popularity in 2023 following the publication of the Stable Diffusion model, capable of text-to-image conversion. Diffusion-based methods have the potential to significantly enhance resistance to adversarial attacks (particularly the increasingly common generative attacks) by embedding watermarks in the latent space.

It is worth noting the decreasing time between the development of a new architecture and its application in watermarking. In the early stages, technologies such as convolutional networks or autoencoders were used for watermarking only several years after their inception.

https://doi.org/10.3390/s26020444

However, with the advancement of deep learning, increased availability of pre-trained models, and improved computational resources, this time has significantly decreased: 4 years for the GAN architecture, and 2 years for Transformers and Swin Transformers. The most dynamic progress can be observed with Stable Diffusion, which was adapted for image watermarking within approximately one year after its release through the ZoDiac architecture [163]. This trend suggests that new technologies are being implemented in watermarking almost immediately after their development, driven by the growing availability of pre-trained models and the increasing demand for more resilient methods to protect digital content.

The evolution of image watermarking methods has led to the development of techniques with diverse properties and applications. Table 2 presents a comparison of key methods described in the previous subsection. Due to significant inconsistencies in experimental conditions across the surveyed studies—including variations in image resolution, watermark capacity, attack types, and evaluation metrics—it is not feasible to provide a standardized quantitative comparison of metrics such as PSNR, SSIM, or BER. As a result, Table 2 focuses on qualitative attributes that remain relatively comparable across works, such as architectural design, general watermark capacity, and robustness to commonly reported attacks.

Table 2. Summary of watermarking algorithms based on deep learning.

Year	Reference	Architecture and Technology	Watermark Capacity	Host Image Resolution	Watermark Robustness
2017	[133]	CNN as autoencoder	64 × 64 pixels	128 × 128 pixels	Noise, cropping, JPEG compression, rotation
2017	[134]	CNN with residual blocks	4096 bits	512 × 512 pixels	Affine transform, cropping, JPEG compression, filtering, rotation, rescaling
2021	[135]	CNN and LSTM for audio mapping	8192 audio samples	128 × 128 pixels	Not implemented
2021	[136]	CNN and fully connected Invariance Layer	32 × 32 pixels/1024 bits	128 × 128 pixels	Noise, cropping, blur, JPEG compression
2023	[137]	CNN with DWT as preprocessing	256 bits	256 × 256 pixels	Noise, dropout, JPEG compression
2023	[138]	CNN with DWT as preprocessing	32 × 32 pixels/1024 bits	512 × 512 pixels	Noise, sharpening, smoothing, dropout, JPEG compression
2022	[139]	CNN autoencoder with DWT and IDWT blocks	50–700 bits	400 × 400 pixels	Perspective warp, motion, blur, noise, color manipulation, JPEG compression
2020	[141]	CNN autoencoder	100 bits	400 × 400 pixels	Perspective warp, camera misalignment, blur, color distortion, noise, JPEG compression
2023	[142]	CNN autoencoder and CNN denoising autoencoder	32 × 32 pixels/1024 bits	128 × 128 pixels	Noise, rotation, JPEG compression
2023	[143]	CNN with MHA in Invariant Domain	8 × 8 pixels/64 bits	128 × 128 pixels	Horizontal flip, blur, solarization, brightness adjustment, contrast variation, hue and saturation modulation
2018	[145]	GAN	30 bits	128 × 128 pixels (training) 512 × 512 pixels (testing)	JPEG compression, blur, cropping, dropout

https://doi.org/10.3390/s26020444

Table 2. *Cont.*

Year	Reference	Architecture and Technology	Watermark Capacity	Host Image Resolution	Watermark Robustness
2019	[146]	GAN with min-max optimization	30 bits	128 × 128 pixels	Cropping, cropout, dropout, blur, JPEG compression and combinations
2020	[147]	GAN	64 bits	512 × 512 pixels	Rotation, JPEG compression, noise, cropping, blur, brightness adjustment
2021	[148]	GAN	64 bits	256 × 256 pixels	JPEG compression, rotation, noise, blur cropping, brightness and contrast adjustment, color inversion
2020	[149]	GAN and IGA	256 bits	256 × 256 pixels	Cropping, dropout, JPEG compression, resizing
2020	[150]	GAN and attention	30 bits	64 × 64 pixels	Cropping, cropout, blur, flip, JPEG compression
2023	[151]	attention module, GAN, feature fusion	30 bits	128 × 128 pixels	Cropping, dropout, blur, JPEG compression, resizing
2024	[152]	GAN-LSTM, Adaptive Gannet Optimization	256 × 256 to 1024 × 1024 pixels	256 × 256 to 1024 × 1024 pixels	Noise, median filtering, blur, JPEG compression, cropping, rotation, scaling
2024	[153]	Transformer and ViT	16-word segments	224 × 224 pixels	JPEG compression, noise, rotation, cropping
2023	[154]	ViT	128 bits	256 × 256 pixels	Noise, median filtering, rotation, scaling
2023	[155]	Transformer with DWT preprocessing	binary image 24 × 24 pixels	96 × 96 pixels	Median and gaussian filtering, noise, SPN, JPEG compression, rotation, cropping, scaling
2023	[156]	Transformer, GAN	36 to 100 bits	128 × 128 pixels	Noise, cropout, dropout, JPEG compression, affine transformation
2023	[157]	Swin Transformer, CNN, MA-FFM, Identity module	64 bits	128 × 128 pixels	Cropping, noise, dropout, gaussian and median filtering, JPEG compression
2024	[158]	Swin Transformer with DCT attention block	64 bits	128 × 128 pixels	Cropout, dropout, rotation, scaling, affine transform
2024	[163]	Stable Diffusion	32 bits	64 × 64 × 4 (latent)	JPEG compression, rotation, noise, blur, generative attacks
2024	[165]	Diffusion Probabilistic Model	16384 bits	128 × 128 pixels	JPEG compression, regeneration attacks
2024	[166]	Diffusion Probabilistic Model	32 bits	512 × 512 pixels	JPEG compression, blur, noise, cropping, brightness adjustment, adaptive attacks

The architectural changes in watermarking methods can be seen not merely as technical refinements, but as reflections of an evolving design philosophy—one that seeks to address the three fundamental challenges of image watermarking: embedding capacity, perceptual transparency, and robustness against attacks. The comparative analysis across CNNs, GANs, Transformers, and diffusion models clearly reflects how different design strategies prioritize or balance these criteria. The first approaches, based on CNNs in encoder–decoder architectures [134] or autoencoders [133] offer simplicity and computational efficiency, but may lack resistance to more complex manipulations. GAN-based solutions improved watermark concealment and realism, at the cost of stability and training complexity. Transformer-based

https://doi.org/10.3390/s26020444

models introduced greater capacity and robustness through attention mechanisms, while diffusion models provide enhanced resilience against generative attacks by embedding in the latent space. With the advancement of technology, the capacity of embedded watermarks has significantly increased, reaching values of bpp = 1 in some methods [165]. Additionally, a growing number of modern watermarking methods are being designed to be resolution-independent, allowing for effective application in HD (high definition) and 4K materials [167,168]. Thus, the architectural perspective adopted in this review implicitly captures the evolution of trade-offs between robustness, invisibility, and capacity—a key concern in watermarking system design.

5. Datasets for Image Watermarking

In principle, any image dataset, whether labeled or unlabeled, can be used to train a watermark embedding and extraction algorithm. However, in the case of deep learning-based methods, the quality and diversity of the training data play a crucial role in determining the model's effectiveness. A well-chosen dataset enables better model generalization, which enhances resistance to attacks and improves watermarking effectiveness under various conditions [169]. Below are the key criteria that a dataset should meet to be suitable for training a watermarking algorithm:

- High resolution—Modern watermarking methods should be tested not only on standard images of 128 × 128 or 256 × 256 pixels but also on high-resolution images such as HD (1080p) and 4K, which is essential for practical applications such as multimedia content protection [170];
- Content diversity—The dataset should include both real-world images (landscapes, faces, animals, objects, vehicles) and graphics or textures. This is particularly important for methods utilizing attention mechanisms, which rely on contextual relationships between image elements [129];
- Open access—Free and open access to data facilitates research replication and the comparison of different solutions' effectiveness, forming the foundation for reliable evaluation of watermarking methods [171];
- High visual quality—Images should be artifact-free, clear, and detailed, allowing for precise evaluation of watermark transparency and its impact on the visual quality of the image after embedding and extraction [172];
- No lossy compression—Lossless formats (e.g., TIFF or PNG) are preferred to avoid artifacts resulting from lossy compression (e.g., JPEG), ensuring a reliable assessment of the watermarking method's resistance to image degradation [173];
- Synthetic and real images—With the growing popularity of generative models such as DALL-E [174] and Midjourney, there is an increasing need to watermark content generated by artificial intelligence. Therefore, the dataset should include both real and synthetic images to ensure algorithm effectiveness in both contexts [175].

Considering the above features when selecting a dataset is crucial for optimizing the effectiveness of watermarking methods and their resistance to various attack scenarios. Based on these criteria, the datasets used can be divided into four categories:

- Benchmark datasets—These are classical image sets widely used in image processing and deep learning research. They are characterized by standard resolutions, usually 256 × 256 pixels, and a high diversity of content, enabling versatile usage;
- High-resolution datasets—These include images with HD and 4K resolutions. Originally intended for training super-resolution algorithms, they are now successfully used to evaluate watermarking effectiveness in real-world applications where high visual quality is essential;

https://doi.org/10.3390/s26020444

- Synthetic datasets—Comprising images generated by AI models, these datasets feature visual characteristics that differ from those of real-world images. As a result, models trained on such datasets may require adapted watermarking methods to perform effectively;
- Specialized datasets—These include images from specific fields, such as medicine, geoinformatics, security, or digital documents, where watermarking plays a key role in ensuring data integrity and authenticity. Such images are characterized by a high level of detail and specific visual features, often necessitating tailored watermarking approaches.

Table 3 presents an overview of the most commonly used datasets in image watermarking research. Key attributes are included, such as resolution, number of images, and thematic content. For each dataset, its main limitations and typical applications in watermarking experiments are also indicated. This aims to facilitate the evaluation of their suitability for various research objectives, such as testing embedding capacity, robustness, or imperceptibility. All listed datasets are publicly available resources.

Table 3. Overview of commonly used datasets in image watermarking research.

Name	Category	Number of Images	Resolution [Pixels]	Type	Limitations/ Notes	Reference
ImageNet	Benchmark	14 million	~224 × 224 to 256 × 256 (resized)	Animals, vehicles, plants, tools	Large scale, requires preprocessing, good for robustness tests.	[176]
COCO	Benchmark	330,000	640 × 480	People, vehicles, food, animals	Moderate resolution, suitable for general-training.	[177]
CIFAR-10/ CIFAR-100	Benchmark	60,000/100,000	32 × 32	Animals, vehicles	Very low resolution, not suitable for perceptual metrics, useful for capacity tests.	[178]
Pascal VOC	Benchmark	21,000	500 × 375	Animals, vehicles, people	Limited scale and resolution, used in simple robustness evaluations.	[179]
BOSSbase	Benchmark	10,000	512 × 512	Grayscale natural images	Designed for steganalysis, great for statistical robustness tests.	[180]
DIV2K	High resolution	1000	~2K (e.g., 2040 × 1080)	Landscape, buildings, architecture,	High quality, ideal for transparency tests.	[181]
Flickr2K	High resolution	2650	~2K (e.g., 2040 × 1350)	Natural photos: portraits, landscapes,	Unprocessed, variable quality, useful for perceptual metrics.	[181]
LIU4K	High resolution	2100	4K (3840 × 2160)	Different background and objects	High resolution, good for visual quality and real-world simulations.	[182]
UHD4K	High resolution	5000+	4K (3840 × 2160)	Satellite images, films, urban scenes	Very high resolution, good for high-end use cases.	[183]
UHD8K	High resolution	2966	8K (7680 × 4320)	Satellite images, films, urban scenes	Extremely high resolution, useful for stress testing.	[183]
LAION-5B	Synthetic	5.85 billion	from 256 × 256 to 4K	Images paired with text prompts (mixture of real and AI-generated)	Unprocessed and noisy, not ideal for reproducible benchmarking.	[184]

https://doi.org/10.3390/s26020444

Table 3. *Cont.*

Name	Category	Number of Images	Resolution [Pixels]	Type	Limitations/ Notes	Reference
CIFAKE	Synthetic	120,000	32 × 32	Real images from CIFAR 10 and synthetic images	Low resolution, designed for deepfake detection benchmarks.	[185]
ArtiFact	Synthetic	1.5 million	from 256 × 256 to 1024 × 1024	People, animals, vehicles, artworks	Moderate resolution, good for testing synthetic distortions.	[186]
ImagiNet	Synthetic	200,000	from 256 × 256 to 2K	Photos, painting	Well-balanced synthetic content, useful for hybrid real/ synthetic training.	[187]
NIH Chest X-ray	Specialized (medical)	112,000	1024 × 1024	Chest X-rays	Suitable for medical robustness/ embedding studies.	[188]
EuroSAT	Specialized (satellite images)	27,000	64 × 64	Satellite images: forests, urban areas, fields	Low resolution, useful for satellite-specific tests.	[189]
LFW (Labeled Faces in the Wild)	Specialized (faces)	13,000	250 × 250	Facial photos in natural conditions	Standard for face datasets, useful for privacy, detection, and watermarking on identity data.	[190]

Most datasets used in image watermarking research were not originally created for watermarking purposes but were primarily developed for classification [176–178,185,189], identification [188,190], or image segmentation [177,179]. Nevertheless, due to their diversity, high quality, and wide availability, they are successfully adapted to evaluate watermarking methods in terms of both resistance to attacks and transparency verification. In particular, benchmark datasets and high-resolution datasets enable the evaluation of watermarking methods under realistic conditions, while synthetic datasets are playing an increasingly significant role in watermarking AI-generated content.

6. New Research Directions and Challenges in Image Watermarking

The dynamic development of deep learning-based watermarking methods has made image watermarking not only more efficient and effective but also more technically and implementationally complex. Traditional approaches based on spatial and frequency domain transforms are gradually losing their effectiveness in the face of the increasing number of generative attacks and the growing need to ensure resistance to various visual distortions. As a result, watermarking is encountering new challenges and development directions, which define current research priorities and determine the future of this technology in the context of digital content protection [191].

6.1. Key Challenges in Implementing Image Watermarking Systems

The practical implementation of deep learning-based image watermarking solutions involves a range of technological, performance, and legal challenges. Methods utilizing artificial intelligence algorithms are significantly more computationally complex than their classical counterparts, which rely on much simpler mathematical algorithms [192]. Their implementation requires advanced models such as autoencoders, convolutional networks, generative networks, or transformers, all characterized by a high number of parameters and the necessity to operate on large datasets [193]. This issue becomes particularly significant

https://doi.org/10.3390/s26020444

when dealing with high-resolution images (e.g., 4K or 8K), where both the embedding and extraction processes demand substantial computational power.

In addition to computational complexity, a fundamental challenge in the practical design of watermarking systems remains the inherent trade-off between three key objectives: robustness, perceptual transparency, and embedding capacity. Achieving a balance between these aspects is non-trivial, as improvements in one dimension often lead to compromises in another. For example, highly robust watermarking methods—especially those based on diffusion models [141]—tend to introduce higher computational costs and longer processing times, while increasing capacity may impact transparency. In particular, diffusion-based approaches, though capable of superior robustness against generative and latent-space attacks, remain difficult to optimize for real-time or resource-constrained applications due to their iterative nature and substantial hardware demands. As a result, ongoing research increasingly focuses not only on model efficiency but also on understanding how architectural choices influence this robustness–transparency–capacity trade-off in practical deployment scenarios.

This trade-off becomes particularly evident in the recent architectures, where excessive computational load and prolonged data processing time constitute a major barrier to real-time deployment. In diffusion-based watermarking methods [194], despite their high robustness and transparency, impose substantial hardware and memory requirements, limiting their applicability in resource-constrained environments.

From a deployment perspective, it is also important to distinguish between training and inference costs in deep learning–based watermarking systems. While training phases—particularly for transformer- and diffusion-based models—are typically performed offline and can leverage high-performance computing resources, inference-time requirements directly affect system feasibility in real-world applications. Large model sizes, memory footprints, and iterative generation processes may limit the applicability of such approaches in real-time, embedded, or resource-constrained environments. Consequently, practical watermarking deployments often prioritize architectures with moderate model size and predictable inference latency, even at the cost of reduced robustness against advanced attacks.

From a quantitative perspective, deep learning architectures used in watermarking differ substantially in scale and computational cost. CNN-based watermarking models typically consist of only a few million parameters and allow single-pass inference, making them suitable for real-time processing on standard GPUs or even high-end embedded devices. In contrast, transformer-based architectures often require tens to hundreds of millions of parameters, while diffusion-based watermarking systems involve iterative generation processes with dozens or hundreds of diffusion steps, resulting in significantly higher inference latency and memory consumption [195]. These order-of-magnitude differences, rather than exact numerical values, are the dominant factor determining deployment feasibility across application scenarios [196].

In the face of increasing computational demands, there is a need to develop techniques that reduce the complexity of deep learning models in the field of watermarking, where embedding effectiveness, processing time, and resource efficiency are crucial. One approach is the use of lightweight architectures designed to require fewer parameters and computational operations while maintaining acceptable processing quality [197]. Examples include models like MobileNet [198] and SqueezeNet [199], which, thanks to a reduced number of filters and innovative convolutional layers, can efficiently operate even on mobile devices.

Another popular strategy is weight pruning, which involves eliminating model parameters that have minimal impact on the output. This reduces the number of insignificant connections in the neural network while maintaining comparable accuracy [200,201]. An alternative method is quantization, which does not eliminate parameters but reduces their bit precision, thereby decreasing model size and accelerating computations [202].

 https://doi.org/10.3390/s26020444

In the context of diffusion models, modifications are gradually emerging to reduce the number of iterations required to achieve high-quality results. Examples include methods that integrate the inference process into the training phase for joint optimization [203] or those aimed at accelerating sampling [204].

Transfer learning is also widely used, allowing previously trained models to be adapted to new domains or tasks, thus shortening training time and reducing data requirements. In diffusion-based learning, pretrained models are utilized to denoise images with embedded watermarks in the latent space, ensuring high transparency and bypassing the costly process of training a model from scratch [163].

Despite the implementation of numerous techniques to limit computational complexity, the scalability of watermarking systems remains a challenge. With the growing number of users, the diversity and volume of image data, and the demand for real-time processing, traditional solutions may prove inadequate [205]. Designing and deploying scalable methods require both suitable hardware architectures and efficient distributed algorithms capable of dynamically managing computational and memory resources [206].

Cloud infrastructure is one of the most popular solutions for improving the scalability of watermarking systems. The cloud enables, among other things, the centralization of deep learning models [207], which significantly simplifies the process of updating and deploying new versions without user-side intervention. Additionally, serverless services allow for automatic scaling in response to changing loads [208].

Another approach is edge computing, which shifts part of the computational operations closer to the data source—directly onto mobile devices or edge servers [209]. This is particularly important for real-time watermarking systems, as it minimizes latency during live transmission, facilitates watermarking of images in IoT systems, and enables fast multimedia content authentication [210]. Edge computing reduces network load since images do not need to be transmitted to a centralized server and also enhances data security [211], by allowing local watermarking without leaving the user's device.

In the training process itself, federated learning is commonly used, enabling deep learning models to be trained using distributed datasets without needing to centralize data on a single server [212]. In watermarking, this approach is especially valuable when data is sensitive or too large to consolidate easily.

From a scalability perspective, federated learning enables efficient utilization of the computational resources of multiple devices simultaneously, reducing reliance on central servers and allowing the training process to scale to thousands of nodes [213]. This enables the parallelization of training process without increasing network load and mitigates bottlenecks associated with transmitting large datasets. An added advantage is the model's adaptability to local data, which can improve watermarking effectiveness in specific conditions and for particular end devices. Federated learning is also used in training diffusion models, significantly reducing the number of parameters while maintaining high output image quality [214].

Ensuring scalability and efficiency in deep learning-based watermarking systems requires a multidimensional approach that combines architecture optimization, effective resource management, and technology adaptation to specific conditions. The solutions described demonstrate significant potential in reducing computational costs, improving response times, and enhancing data processing security. However, their effectiveness depends on multiple factors, with the greatest efficiency often achieved through a synergistic combination of several techniques. Integrated approaches like Edge Intelligence [215], which combine AI, edge computing, and federated learning, are already opening new possibilities for broad watermarking applications in resource-constrained environments and real-time systems.

https://doi.org/10.3390/s26020444

6.2. Research Directions for Methods and Algorithms

The dynamic development of deep learning in watermarking opens new research directions focused on designing architectures aimed at improving the robustness, efficiency, and transparency of watermarking systems. As shown by the literature review conducted in Chapter 4, two groups of models have begun to dominate recent research: Transformers (Vision Transformer and Swin Transformer) and diffusion models. Both approaches offer an intriguing alternative to previously used solutions, namelyCNNs [216] and generative adversarial networks (GANs) [217].

From a critical standpoint, recent advances in deep learning-based watermarking reveal that the choice of network architecture directly determines not only robustness and transparency but also vulnerability to specific classes of attacks and the feasibility of real-world deployment. Consequently, CNN-, GAN-, Transformer-, and diffusion-based approaches should be viewed as complementary rather than competing solutions, each addressing different design constraints and threat models.

Transformer-based architectures, utilizing various attention mechanism variants, are increasingly applied to diverse image processing tasks [218]. ViT and its improved variant, Swin Transformer, introduce fundamental changes to how images are processed. Unlike traditional CNNs, which analyze images locally, Transformers capture global dependencies between different parts of an image. Swin Transformers further enhance this by introducing a sliding window mechanism, enabling hierarchical image processing and improving both computational efficiency and the ability to analyze local and global features [159].

Diffusion models are generative probabilistic models where image generation (or other processing operations) is performed through iterative noise addition and removal. Unlike GAN models, which often face convergence issues, diffusion model training is much more stable and yields higher quality generated images [115,125,219].

Tables 4 and 5 present a comparative overview of the key features of CNNs and transformer-based architectures, as well as diffusion models and GANs, with a focus on their applications in image watermarking.

Table 4. Comparison of key features of Vision Transformer, Swin Transformer, and CNN in the context of watermarking.

Feature	CNN	ViT	Swin Transformer
Feature processing method	Local (by convolutional filters)	Global (by self-attention)	Local and global (by shifted windows)
Resistance to traditional attacks	Medium	High	High
Resistance to generative attacks	Medium	High	Very high
Computational complexity	Low to medium	High	Medium (optimized)
Ability to capture context	Limited	High	High (with local optimization)
Scalability to high resolution	Limited	Limited (without optimization)	High
Potential in watermarking	Well verified but limited	High (based on previous research)	Very high

Table 5. Comparison of diffusion models and generative networks (GANs) in the context of watermarking.

Feature	GAN	Diffusion Model
Training stability	Low (frequent convergence problems)	High
Quality of generated images	High	Very high
Generation time	Relatively short	Longer (if no optimization methods were used)
Resistance to attacks	Medium	High
Computational complexity	Medium	High
Possibility of embedding in latent space	Limited	Yes
Potential in watermarking	Well verified but limited	High (based on previous research)

https://doi.org/10.3390/s26020444

The analysis of the presented data indicates that further research on visual transformers in the context of watermarking should primarily focus on optimizing computational complexity to enhance their potential in real-time systems. Although architectures such as ViT and the Swin Transformer demonstrate high effectiveness in watermark embedding [153,154,157,158], their intensive computational requirements limit practical deployment possibilities. Therefore, an important research direction is the development of lightweight and more efficient versions of transformers capable of effectively operating on end devices such as smartphones or surveillance cameras [220–222].

Due to the simplicity and good efficiency of CNNs, combining transformers with convolutional networks in the form of hybrid architectures is a valuable approach. These models, merging the local precision of CNNs with the global context of transformers, can improve watermark durability and resilience while maintaining an acceptable level of computational complexity [223]. With the growing popularity of generative attacks, research should also focus on increasing watermark resistance to such threats [224]. A significant challenge lies in better understanding how the self-attention mechanism affects the placement and durability of the watermark in the image and how it can be used to embed better-hidden yet higher-capacity watermarks [225].

For diffusion models, research mainly focuses on developing technologies that reduce the number of diffusion steps without degrading the quality of the generated images [226]. Traditional models, such as Denoising Diffusion Probabilistic Model or Stable Diffusion, while providing high-quality watermark embedding [163,165,166], are characterized by long generation times, hindering their application in real-time and low-latency environments. In response, efforts are being made to develop accelerated sampling methods and to utilize latent diffusion models, which operate in the latent space [227], significantly shortening watermark embedding time. This approach not only improves watermark resistance against subsequent manipulations but also enables more controlled and precise digital content watermarking [164].

Research in this area also focuses on how the watermark can survive various transformations, such as lossy compression, scaling, and attacks using generative models, which are becoming increasingly common. Additionally, a key development direction involves designing hybrid architectures combining diffusion models with transformers, leveraging the global understanding of images provided by the self-attention mechanism and the stable generation process offered by diffusion [228,229]. Researchers are also exploring methods combining modern neural network architectures with quantum technology [230].

Taken together, the comparisons presented in Tables 4 and 5 highlight a fundamental trade-off in deep learning–based watermarking. Architectures that provide the highest robustness against generative and latent-space attacks—such as diffusion models and large transformer-based solutions- are currently the most computationally demanding and the least suitable for real-time or resource-constrained deployment scenarios. In contrast, CNN-based and hybrid CNN-Transformer approaches, while offering lower resistance to advanced generative attacks, remain more practical for time-critical and embedded applications due to their reduced computational complexity.

Overall, current research clearly indicates that no single deep learning architecture can simultaneously satisfy all watermarking requirements, including robustness, transparency, capacity, and computational efficiency. As a result, future research increasingly points toward adaptive and hybrid solutions, in which architectural choices are guided by specific application scenarios, threat models, and deployment constraints rather than by the pursuit of a universal, one-size-fits-all watermarking framework.

https://doi.org/10.3390/s26020444

6.3. Application-Oriented Research in Watermarking

Contemporary watermarking is increasingly expanding beyond traditional intellectual property protection [231]. With the use of deep learning, modern watermarking systems are being applied in many key areas, including the identification of AI-generated content, cybersecurity, monitoring systems, and IoT. Below are the current and future applications of watermarking that may dominate this field in the coming years.

6.3.1. Watermarking in Identification of Deepfakes

The rapid development of generative techniques has made the identification and counteraction of so-called deepfakes a key challenge in digital security [232]. Fake multimedia content, generated by artificial intelligence, is becoming increasingly realistic and difficult to distinguish from authentic materials, posing serious threats in the realms of politics, security, and privacy [233].

The use of watermarking techniques is playing an increasingly important role in the recognition of AI-generated content [234], including deepfakes detection. Watermarking algorithms can be employed to embed invisible markers during the image generation process, enabling subsequent verification of whether a given material was synthetically generated. An example of such an approach is SynthID [235], developed by Google, which integrates invisible watermarks into AI-generated content to facilitate their identification. However, ensuring the resilience of these markers against attempts at removal or modification is crucial for effectively combating disinformation on the Internet. Research must also consider techniques that are resistant to common deepfake attacks and operations, such as face alterations, voice modulation, or background modifications.

An important emerging research area is the embedding of metadata in deepfakes. By embedding source information or generation timestamps directly into multimedia content, it becomes possible to detect manipulations and track the origin of materials. Standards such as the Content Authenticity Initiative (CAI) and the Coalition for Content Provenance and Authenticity (C2PA) promote this approach, enabling the embedding of metadata in images and other media to verify their authenticity. The implementation of such standards can significantly aid in the fight against disinformation and facilitate the identification of AI-generated content.

6.3.2. Watermarking in Cybersecurity

In the context of cybersecurity, there is a growing number of applications utilizing watermarking technologies in various ways—often referred to as Cyber Watermarking [236]. A highly promising development direction is the integration of modern watermarking techniques with blockchain technology [237]. Combining these two solutions enables the creation of systems that not only mark digital content but also store cryptographic hashes of the content in a decentralized database [238]. This allows any user to verify the authenticity of multimedia files without concerns about unauthorized manipulation within the chain.

Watermarking is also gaining popularity as an authentication mechanism in both commercial and governmental applications. Embedding digital signatures into images and other documents can serve to confirm their authenticity and prevent unauthorized modifications [239]. Research in this area mainly focuses on increasing watermark resilience against compression, format conversions, and content edits that may occur during data transmission or archiving.

https://doi.org/10.3390/s26020444

6.3.3. Watermarking in Monitoring and IoT Systems

The dynamic development of Internet of Things technologies and video surveillance systems has opened new perspectives for the application of watermarking, extending beyond traditional multimedia content protection. Modern surveillance systems, integral to smart cities and advanced security infrastructures, rely on the credibility of collected video data. In this context, watermarking algorithms play a crucial role in ensuring the integrity and authenticity of recordings, enabling rapid identification of visual materials and detection of manipulation attempts [240].

One of the most important applications of watermarking in surveillance systems is the verification of recording authenticity for criminal investigations and legal proceedings. Recordings containing an embedded watermark can serve as reliable evidence in court cases, confirming that the material has not been altered since its capture. When data is transmitted over networks or stored long-term, watermarks enable material tracking, allowing any attempt to replace or delete portions of the footage to be quickly detected.

Beyond traditional surveillance systems, watermarking is gaining significance in IoT ecosystems, which include a wide range of image-capturing devices such as drones, wearable cameras, and systems in autonomous vehicles [241]. In such cases, watermarking serves not only to identify the data source and control access but also to verify recording authenticity in real time.

One of the key challenges associated with the use of watermarking in monitoring systems is ensuring resistance to varying environmental conditions. Recordings can be susceptible to distortions caused by lighting changes, weather conditions, or recording angles.

7. Conclusions

The rapid development of deep learning technologies in recent years has significantly impacted the field of digital image watermarking, offering new possibilities for improving transparency, robustness, and embedding capacity of watermarking systems. This article provides a comprehensive review of both traditional watermarking methods and the latest deep learning-based approaches, covering architectures such as convolutional neural networks, generative adversarial networks, Vision Transformers, Swin Transformers, and diffusion models. The conducted analysis shows that, while deep learning-based methods offer significant advantages in terms of embedding effectiveness and attack resistance, they also present challenges related to high computational requirements, the need for large datasets, and architectural complexity.

The analysis indicates that the latest architectures —particularly transformers and diffusion models, offer a promising balance between transparency and robustness in watermarking systems. Compared to older neural network architectures, these models can effectively conceal watermarks without noticeably degrading image quality, while demonstrating high resilience against generative attacks, including those performed using advanced deep learning models. This is particularly important in the context of rapidly developing generative technologies, which pose an increasing threat to the integrity of digital content. These trends indicate that transformers and diffusion models playing an increasingly significant role in watermarking research in the coming years. However, it is important to emphasize that other architectures, such as CNNs and GANs, will not be entirely phased out. On the contrary, there is already a significant rise in the popularity of hybrid solutions that combine the strengths of different models—for example, systems integrating transformers with convolutional networks or transformers with diffusion models. Additionally, there is growing interest in supporting classical neural methods with

https://doi.org/10.3390/s26020444

quantum technologies, which may open new perspectives for enhancing performance and increasing the robustness of watermarking systems.

Overall, while deep learning-based watermarking methods have greatly expanded the possibilities for digital content protection, the challenges associated with their practical implementation and the dynamic evolution of generative technologies make further research in this area essential. Moreover, technological changes and an expanding range of applications clearly signal a sift away from an era in which watermarking was primarily associated with copyright protection. With the rapid development of new use cases—such as cybersecurity, Internet of Things (IoT), authentication systems, and multimedia forgery detection—watermarking is becoming crucial for ensuring data integrity, authenticity, and security. Thanks to these new possibilities, watermarking could evolve into a fundamental component of authorization systems and become one of the main tools for content origin identification and credibility verification in the digital ecosystem of the future.

Author Contributions: Conceptualization, M.B. and Z.P.; methodology, M.B.; validation, J.M.Ż. and Z.P.; formal analysis, M.B.; resources, M.B.; writing—original draft preparation, M.B.; writing—review and editing, Z.P. and J.M.Ż.; visualization, M.B.; supervision, Z.P. and J.M.Ż.; project administration, Z.P.; funding acquisition, Z.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Military University of Technology, Faculty of Electronics, grant number UGB 22-054/2025, titled: "New Neural Network Architectures for Signal and Data Processing in Radiocommunications and Multimedia".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created in this study. This survey analyzes and reviews publicly available datasets commonly used in image watermarking research, including ImageNet, COCO, CIFAR-10/100, Pascal VOC, BOSSBase, DIV2K, Flickr2K, LAION-5B, CIFAKE, ArtiFact, ImagiNet, NIH Chest X-ray, EuroSAT, and LFW. All datasets referenced in this article are openly accessible through their respective repositories as cited in the manuscript. Detailed dataset descriptions, resolutions, and usage contexts are provided in Table 3.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
GAN	Generative Adversarial Network
IoT	Internet of Things
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
ViT	Vision Transformer
VIF	Visual Information Fidelity

References

1. Spaiser, V. Chapter 19: Digital Data and Methods. In *Sociology, Social Policy and Education*; Elgar Publishing: Cheltenham, UK, 2021; ISBN 978-1-78990-685-1.

https://doi.org/10.3390/s26020444

2. Seetharamu, S.; Cn, L.M.; Bhattacharya, A.; Bt, D.C. Digital Data Protection Laws: A Review. *Int. J. Sci. Res. Sci. Eng. Technol.* **2024**, *11*, 64–75. [CrossRef]
3. Czetwertyński, S. Digital Piracy: The Issue of Knowledge of the Institution of Copyright Law. *Ekon. Prawo Econ. Law* **2023**, *22*, 69–102. [CrossRef]
4. Townsened, C. The Consequences of Digital Piracy. Available online: https://www.uscybersecurity.net/digital-piracy/ (accessed on 7 October 2024).
5. Sadiku, M.N.O.; Ashaolu, T.J.; Ajayi-Majebi, A.; Musa, S.M. Digital Piracy. *IJSCIA* **2021**, *2*, 797–800. [CrossRef]
6. Javed, A.R.; Jalil, Z.; Zehra, W.; Gadekallu, T.R.; Suh, D.Y.; Piran, M.J. A Comprehensive Survey on Digital Video Forensics: Taxonomy, Challenges, and Future Directions. *Eng. Appl. Artif. Intell.* **2021**, *106*, 104456. [CrossRef]
7. Walczyna, T.; Piotrowski, Z. Quick Overview of Face Swap Deep Fakes. *Appl. Sci.* **2023**, *13*, 6711. [CrossRef]
8. Petitcolas, F.; Anderson, R.; Kuhn, M. Information Hiding—A Survey. *Proc. IEEE* **1999**, *87*, 1062–1078. [CrossRef]
9. Cheddad, A.; Condell, J.; Curran, K.; Mc Kevitt, P. Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Process.* **2010**, *90*, 727–752. [CrossRef]
10. Jekateryńczuk, G.; Jankowski, D.; Veyland, R.; Piotrowski, Z. Detecting Malicious Devices in IPSEC Traffic with IPv4 Steganography. *Appl. Sci.* **2024**, *14*, 3934. [CrossRef]
11. Rocha, A.; Goldenstein, S. Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah? *Rev. Informática Teórica Apl.* **2008**, *15*, 83–110. [CrossRef]
12. Zhang, Y. Digital Watermarking Technology: A Review. In Proceedings of the 2009 ETP International Conference on Future Computer and Communication, Wuhan, China, 6–7 June 2009; pp. 250–252.
13. Bailey, J. Watermarking vs. Fingerprinting: A War in Terminology. Plagiarism Today 2007. Available online: https://www.plagiarismtoday.com/2007/10/09/watermarking-vs-fingerprinting-a-war-in-terminology/ (accessed on 6 November 2025).
14. Hosny, K.M.; Magdi, A.; ElKomy, O.; Hamza, H.M. Digital Image Watermarking Using Deep Learning: A Survey. *Comput. Sci. Rev.* **2024**, *53*, 100662. [CrossRef]
15. Hu, K.; Wang, M.; Ma, X.; Chen, J.; Wang, X.; Wang, X. Learning-Based Image Steganography and Watermarking: A Survey. *Expert Syst. Appl.* **2024**, *249*, 123715. [CrossRef]
16. Piotrowski, Z. Drift Correction Modulation Scheme for Digital Signal Processing. *Math. Comput. Model.* **2013**, *57*, 2660–2670. [CrossRef]
17. Piotrowski, Z. Angle Phase Drift Correction Method Effectiveness. In Proceedings of the Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2009, Poznan, Poland, 24–26 September 2009; pp. 82–86.
18. Piotrowski, Z.; Lenarczyk, P. Blind Image Counterwatermarking—Hidden Data Filter. *Multimed. Tools Appl.* **2017**, *76*, 10119–10131. [CrossRef]
19. Hua, G.; Huang, J.; Shi, Y.Q.; Goh, J.; Thing, V.L.L. Twenty Years of Digital Audio Watermarking—A Comprehensive Review. *Signal Process.* **2016**, *128*, 222–242. [CrossRef]
20. Uddin, M.S.; Ohidujjaman; Hasan, M.; Shimamura, T. Audio Watermarking: A Comprehensive Review. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 1410–1418. [CrossRef]
21. Shelke, R.D.; Nemade, M.U. Audio Watermarking Techniques for Copyright Protection: A Review. In Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016; pp. 634–640.
22. Al-khafaji, A.; Alwan; Nur, N.; Sjarif, N.N.A. Digital Text Watermarking Techniques Classification and Open Research Challenges: A Review. *Technol. Rep. Kansai Univ.* **2020**, *62*, 18.
23. Kaur, M.; Mahajan, K. An Existential Review on Text Watermarking Techniques. *Int. J. Comput. Appl.* **2015**, *120*, 29–32. [CrossRef]
24. Kamaruddin, N.S.; Kamsin, A.; Por, L.Y.; Rahman, H. A Review of Text Watermarking: Theory, Methods, and Applications. *IEEE Access* **2018**, *6*, 8011–8028. [CrossRef]
25. Begum, M.; Uddin, M.S. Digital Image Watermarking Techniques: A Review. *Information* **2020**, *11*, 110. [CrossRef]
26. Wadhera, S.; Kamra, D.; Rajpal, A.; Jain, A.; Jain, V. A Comprehensive Review on Digital Image Watermarking. In Proceedings of the 5th International Conference on Computing Sciences (ICCS 2021), Beijing China, 4–6 December 2021.
27. Sharma, S.; Zou, J.J.; Fang, G.; Shukla, P.; Cai, W. A Review of Image Watermarking for Identity Protection and Verification. *Multimed. Tools Appl.* **2024**, *83*, 31829–31891. [CrossRef]
28. Joseph, I.; Mandala, J. Comprehensive Review on Video Watermarking Security Threats, Challenges, and Its Applications. *ECS Trans.* **2022**, *107*, 13833. Available online: https://iopscience.iop.org/article/10.1149/10701.13833ecst (accessed on 10 October 2024). [CrossRef]
29. Paul, R.T. Review of Robust Video Watermarking Techniques. *IJCA Spec. Issue Comput. Sci.* **2011**, *3*, 90–95.

https://doi.org/10.3390/s26020444

30. Dey, A.; Bhattacharya, S.; Chaki, N. Software Watermarking: Progress and Challenges. *INAE Lett.* **2019**, *4*, 65–75. [CrossRef]
31. Dalla Preda, M.; Pasqua, M. Software Watermarking: A Semantics-Based Approach. *Electron. Notes Theor. Comput. Sci.* **2017**, *331*, 71–85. [CrossRef]
32. Li, Y. Database Watermarking: A Systematic View. In *Handbook of Database Security: Applications and Trends*; Gertz, M., Jajodia, S., Eds.; Springer: Boston, MA, USA, 2008; pp. 329–355. ISBN 978-0-387-48533-1.
33. Alqassab, A.; Alanezi, M. Relational Database Watermarking Techniques: A Survey. *J. Phys. Conf. Ser.* **2021**, *1818*, 012185. [CrossRef]
34. Boenisch, F. A Systematic Review on Model Watermarking for Neural Networks. *Front. Big Data* **2021**, *4*, 729663. [CrossRef] [PubMed]
35. Li, Y.; Wang, H.; Barni, M. A Survey of Deep Neural Network Watermarking Techniques. *Neurocomputing* **2021**, *461*, 171–193. [CrossRef]
36. Vasiljević, I.; Obradović, R.; Đurić, I.; Popkonstantinović, B.; Budak, I.; Kulić, L.; Milojević, Z.; Vasiljević, I.; Obradović, R.; Đurić, I.; et al. Copyright Protection of 3D Digitized Artistic Sculptures by Adding Unique Local Inconspicuous Errors by Sculptors. *Appl. Sci.* **2021**, *11*, 7481. [CrossRef]
37. Huang, T.; Huang, J.; Pang, Y.; Yan, H. Smart Contract Watermarking Based on Code Obfuscation. *Inf. Sci.* **2023**, *628*, 439–448. [CrossRef]
38. Li, L.; Jiang, B.; Wang, P.; Ren, K.; Yan, H.; Qiu, X. Watermarking LLMs with Weight Quantization. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023.
39. Rathi, P.; Bhadauria, S.; Rathi, S. Watermarking of Deep Recurrent Neural Network Using Adversarial Examples to Protect Intellectual Property. *Appl. Artif. Intell.* **2022**, *36*, 2008613. [CrossRef]
40. Lenarczyk, P.; Piotrowski, Z. Parallel Blind Digital Image Watermarking in Spatial and Frequency Domains. *Telecommun. Syst.* **2013**, *54*, 287–303. [CrossRef]
41. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
42. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multiscale Structural Similarity for Image Quality Assessment. In Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 9–12 November 2003; pp. 1398–1402.
43. *Recommendation ITU-R BT.500-13*; Methodology for the Subjective Assessment of the Quality of Television Pictures. ITU-R: Geneva, Switzerland, 2023.
44. *Recommendation ITU-T P.910 (10/2023)*; Subjective Video Quality Assessment Methods for Multimedia Applications. ITU-R: Geneva, Switzerland, 2023. Available online: https://www.itu.int/epublications/publication/itu-t-p-910-2023-10-subjective-video-quality-assessment-methods-for-multimedia-applications (accessed on 19 November 2025).
45. Vallez, C.; Kucharavy, A.; Dolamic, L. Needle In A Haystack, Fast: Benchmarking Image Perceptual Similarity Metrics At Scale. *arXiv* **2022**, arXiv:2206.00282. [CrossRef]
46. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
47. Ding, K.; Ma, K.; Wang, S.; Simoncelli, E.P. Image Quality Assessment: Unifying Structure and Texture Similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2567–2581. [CrossRef] [PubMed]
48. Prashnani, E.; Cai, H.; Mostofi, Y.; Sen, P. PieAPP: Perceptual Image-Error Assessment through Pairwise Preference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
49. Wu, J.; Zhou, M.; Zhu, C.; Liu, Y.; Harandi, M.; Li, L. Performance Evaluation of Adversarial Attacks: Discrepancies and Solutions. *arXiv* **2021**, arXiv:2104.11103. [CrossRef]
50. Buzhinsky, I.; Nerinovsky, A.; Tripakis, S. Metrics and Methods for Robustness Evaluation of Neural Networks with Generative Models. *arXiv* **2020**, arXiv:2003.01993. [CrossRef]
51. Rabhi, M.; Pietro, R.D. Adversarial Attacks Neutralization via Data Set Randomization. *arXiv* **2023**, arXiv:2306.12161. [CrossRef]
52. Fang, H.; Chen, K.; Qiu, Y.; Ma, Z.; Zhang, W.; Chang, E.-C. DERO: Diffusion-Model-Erasure Robust Watermarking. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, VIC, Australia, 28 October–1 November 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2973–2981.
53. Guo, Y.; Li, R.; Hui, M.; Guo, H.; Zhang, C.; Cai, C.; Wan, L.; Wang, S. FreqMark: Invisible Image Watermarking via Frequency Based Optimization in Latent Space. *arXiv* **2024**, arXiv:2410.20824. [CrossRef]
54. Cox, I.J. *Digital Watermarking and Steganography*, 2nd ed.; The Morgan Kaufmann Series in Multimedia Information and Systems; Morgan Kaufmann Publishers: Amsterdam, The Netherlands; Boston, MA, USA, 2008; ISBN 978-0-12-372585-1.

55. Tuner, L.F. Digital Data Security System. Patent IPN WO 89/08915, 21 September 1989.
56. van Schyndel, R.G.; Tirkel, A.Z.; Osborne, C.F. A Digital Watermark. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 2, pp. 86–90.
57. Bamatraf, A.; Ibrahim, R.; Salleh, M. Digital Watermarking Algorithm Using LSB. In Proceedings of the 2010 International Conference on Computer Applications and Industrial Electronics, Kuala Lumpur, Malaysia, 5–8 December 2010; p. 159.
58. Bamatraf, A.; Ibrahim, R.; Salleh, M.N.M. A New Digital Watermarking Algorithm Using Combination of Least Significant Bit (LSB) and Inverse Bit. *arXiv* **2011**, arXiv:1111.6727. [CrossRef]
59. Arya, R.K.; Saharan, R. Algorithm to Enhance the Robustness and Imperceptibility of LSB. In Proceedings of the 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, India, 1–2 May 2015; pp. 583–587.
60. Muyco, S.D.; Hernandez, A.A. Least Significant Bit Hash Algorithm for Digital Image Watermarking Authentication. In *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 150–154.
61. Faheem, Z.B.; Ishaq, A.; Rustam, F.; de la Torre Díez, I.; Gavilanes, D.; Vergara, M.M.; Ashraf, I. Image Watermarking Using Least Significant Bit and Canny Edge Detection. *Sensors* **2023**, *23*, 1210. [CrossRef]
62. Yasin, H.M.; Sallow, A.B.; Mahmood, R.Z. High-Speed FPGA-Based Video Watermarking Using LSB Technique in the Spatial Domain. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 644–653.
63. Kadhim, I.J.; Premaratne, P.; Alwan, Z.A. Invisible Video Watermarking Based on Frame Selection and LSB Embedding. *Int. J. Intell. Eng. Syst.* **2024**, *17*, 41–53. [CrossRef]
64. Gottimukkala, A.R.; Kumar, N.; Dash, J.K.; Swain, G. Image Watermarking Based on Remainder Value Differencing and Extended Hamming Code. *J. Electron. Imaging* **2023**, *33*, 011003. [CrossRef]
65. Venugopala, P.S.; Sarojadevi, H.; Chiplunkar, N.N.; Bhat, V. Video Watermarking by Adjusting the Pixel Values and Using Scene Change Detection. In Proceedings of the 2014 Fifth International Conference on Signal and Image Processing, Bangalore, India, 8–10 January 2014; pp. 259–264.
66. Yeo, I.-K.; Kim, H.J. Generalized Patchwork Algorithm for Image Watermarking. *Multimed. Syst.* **2003**, *9*, 261–265. [CrossRef]
67. Purna Kumari, B.; Subramanyam Rallabandi, V.P. Modified Patchwork-Based Watermarking Scheme for Satellite Imagery. *Signal Process.* **2008**, *88*, 891–904. [CrossRef]
68. Kang, H.; Yamaguchi, K.; Kurkoski, B.; Yamaguchi, K. Psychoacoustically-Adapted Patchwork Algorithm for Watermarking. In Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), Kaohsiung, Taiwan, 26–28 November 2007; pp. 267–270.
69. Ali, M. Robust Image Watermarking in Spatial Domain Utilizing Features Equivalent to SVD Transform. *Appl. Sci.* **2023**, *13*, 6105. [CrossRef]
70. Wang, T. Digital Image Watermarking Using Dual-Scrambling and Singular Value Decomposition. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 1, pp. 724–727.
71. Wang, H.; Hao, J.; Cui, F. Colour Image Watermarking Algorithm Based on the Arnold Transform. In Proceedings of the 2010 International Conference on Communications and Mobile Computing, Shenzhen, China, 12–14 April 2010; IEEE Computer Society: Washington, DC, USA, 2010; pp. 66–69.
72. Li, X.; Wang, X.; Yang, W.; Wang, X. A Robust Video Watermarking Scheme to Scalable Recompression and Transcoding. In Proceedings of the 2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 June 2016; pp. 257–260.
73. Li, X.; Wang, X.; Chen, A.; Xiao, L. A Simplified and Robust DCT-Based Watermarking Algorithm. In Proceedings of the 2017 2nd International Conference on Multimedia and Image Processing (ICMIP), Wuhan, China, 17–19 March 2017; pp. 167–171.
74. Moosazadeh, M.; Ekbatanifard, G. An Improved Robust Image Watermarking Method Using DCT and YCoCg-R Color Space. *Optik* **2017**, *140*, 975–988. [CrossRef]
75. Badran, E.F.; Ghobashy, A.; El-Shennawy, K. DCT-Based Digital Image Watermarking Via Image Segmentation Techniques. In Proceedings of the 2006 ITI 4th International Conference on Information & Communications Technology, Cairo, Egypt, 10–12 December 2006; p. 1.
76. Alomoush, W.; Khashan, O.A.; Alrosan, A.; Attar, H.H.; Almomani, A.; Alhosban, F.; Makhadmeh, S.N. Digital Image Watermarking Using Discrete Cosine Transformation Based Linear Modulation. *J. Cloud Comput.* **2023**, *12*, 96. [CrossRef]
77. Li, H.; Guo, X. Embedding and Extracting Digital Watermark Based on DCT Algorithm. *J. Comput. Commun.* **2018**, *6*, 287–298. [CrossRef]

https://doi.org/10.3390/s26020444

78. Xu, Z.J.; Wang, Z.Z.; Lu, Q. Research on Image Watermarking Algorithm Based on DCT. *Procedia Environ. Sci.* **2011**, *10*, 1129–1135. [CrossRef]
79. Senthilkumaran, N.; Abinaya, S. Digital Image Watermarking Using Dft Algorithm. *Adv. Comput. Int. J.* **2016**, *7*, 9–17. [CrossRef]
80. Cedillo-Hernandez, M.; Cedillo-Hernandez, A.; Garcia-Ugalde, F.J. Improving DFT-Based Image Watermarking Using Particle Swarm Optimization Algorithm. *Mathematics* **2021**, *9*, 1795. [CrossRef]
81. Jimson, N.; Hemachandran, K. DFT Based Digital Image Watermarking: A Survey. *Int. J. Adv. Res. Comput. Sci.* **2018**, *9*, 540–544. [CrossRef]
82. Pun, C. A Novel DFT-Based Digital Watermarking System for Images. In Proceedings of the 2006 8th international Conference on Signal Processing, Guilin, China, 16–20 November 2006; Volume 2.
83. Malonia, M.; Agarwal, S.K. Digital Image Watermarking Using Discrete Wavelet Transform and Arithmetic Progression Technique. In Proceedings of the 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 5–6 March 2016; pp. 1–6.
84. Begum, M.; Shorif, S.B.; Uddin, M.S.; Ferdush, J.; Jan, T.; Barros, A.; Whaiduzzaman, M. Image Watermarking Using Discrete Wavelet Transform and Singular Value Decomposition for Enhanced Imperceptibility and Robustness. *Algorithms* **2024**, *17*, 32. [CrossRef]
85. Giri, K.J.; Quadri, S.M.K.; Bashir, R.; Bhat, J.I. DWT Based Color Image Watermarking: A Review. *Multimed. Tools Appl.* **2020**, *79*, 32881–32895. [CrossRef]
86. Ramos, A.M.; Artiles, J.A.P.; Chaves, D.P.B.; Pimentel, C. A Fragile Image Watermarking Scheme in DWT Domain Using Chaotic Sequences and Error-Correcting Codes. *Entropy* **2023**, *25*, 508. [CrossRef]
87. Al-Haj, A. Combined DWT-DCT Digital Image Watermarking. *J. Comput. Sci.* **2007**, *3*, 740–746. [CrossRef]
88. Fazli, S.; Moeini, M. A Robust Image Watermarking Method Based on DWT, DCT, and SVD Using a New Technique for Correction of Main Geometric Attacks. *Optik* **2016**, *127*, 964–972. [CrossRef]
89. Hamidi, M.; El Haziti, M.; Cherifi, H.; El Hassouni, M. A Hybrid Robust Image Watermarking Method Based on DWT-DCT and SIFT for Copyright Protection. *J. Imaging* **2021**, *7*, 218. [CrossRef]
90. Amirgholipour Kasmani, S.; Naghsh-Nilchi, A. Robust Digital Image Watermarking Based on Joint DWT-DCT. *JDCTA* **2009**, *3*, 42–54. [CrossRef]
91. Radhi, H.Y.; Yousif, S.F.; Mohamed, W.Q.; Mohammed, A.H. Efficient Image Watermark Scheme Based on DCT-DFT and Singular Value Decomposition Algorithm with Ikeda/Arnold Maps. *J. Wuhan Univ. Technol. Mater. Sci. Ed.* **2023**, *47*, 885–898.
92. Li, L.; Zhang, H.-J.; Fan, H.-Y.; Lu, Z.-M. A DFT and IWT-DCT Based Image Watermarking Scheme for Industry. *IEICE Trans. Inf. Syst.* **2023**, *106*, 1916–1921. [CrossRef]
93. Varghese, J.; Bin Hussain, O.; Subash, S.; Razak, A. An Effective Digital Image Watermarking Scheme Incorporating DCT, DFT and SVD Transformations. *PeerJ Comput. Sci.* **2023**, *9*, e1427. [CrossRef]
94. Ansari, R.; Devanalamath, M.M.; Manikantan, K.; Ramachandran, S. Robust Digital Image Watermarking Algorithm in DWT-DFT-SVD Domain for Color Images. In Proceedings of the 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, 19–20 October 2012; IEEE Conference Publication; IEEE Xplore. Available online: https://ieeexplore.ieee.org/document/6398160 (accessed on 15 October 2024).
95. Chen, A.; Wang, X. An Image Watermarking Scheme Based on DWT and DFT. In Proceedings of the 2017 2nd International Conference on Multimedia and Image Processing (ICMIP), Wuhan, China, 17–19 March 2017; pp. 177–180.
96. Fullea, E.; Mart"nez, J.M. Robust Digital Image Watermarking Using DWT, DFT and Quality Based Average. In Proceedings of the Ninth ACM International Conference on Multimedia, Ottawa, ON, Canada, 30 September–5 October 2001; pp. 489–491.
97. Varghese, J.; Hussain, O.B.; Razak, T.A.; Subash, S. A Hybrid Digital Image Watermarking Scheme Incorporating DWT, DFT, DCT and SVD Transformations. *J. Eng. Res.* **2022**, *10*, 113–130. [CrossRef]
98. Li, J.; Sui, A. A Digital Video Watermarking Algorithm Based on DCT Domain. In Proceedings of the 2012 Fifth International Joint Conference on Computational Sciences and Optimization, Harbin, China, 23–26 June 2012; pp. 557–560.
99. Salih, M.; Ahmed, E.; Abdulkareem Al-Abaji, M. Digital Video Watermarking Methods Using DCT and SVD. *J. Educ. Sci.* **2019**, *29*, 266–278. [CrossRef]
100. Al-Gindy, A.; Omar, A.A.-C.; Mashal, O.; Shaker, Y.; Alhogaraty, E.; Moussa, S. A New Watermarking Scheme for Digital Videos Using DCT. *Open Comput. Sci.* **2022**, *12*, 248–259. [CrossRef]
101. Yang, C.-H.; Huang, H.-Y.; Hsu, W.-H. An Adaptive Video Watermarking Technique Based on DCT Domain. In Proceedings of the 2008 8th IEEE International Conference on Computer and Information Technology, Sydney, NSW, Australia, 8–11 July 2008; pp. 589–594.
102. Hu, J.C.; Huang, H.Y.; Liang, A.H. Video Watermarking Algorithm Based on DCT and SVD. *Appl. Mech. Mater.* **2013**, *321–324*, 2688–2692. [CrossRef]

 https://doi.org/10.3390/s26020444

103. Rajab, L.; Al-Khatib, T.; Al-Haj, A. A Blind DWT-SCHUR Based Digital Video Watermarking Technique. *J. Softw. Eng. Appl.* **2015**, *8*, 224–233. Available online: https://www.scirp.org/journal/paperinformation?paperid=55955 (accessed on 15 October 2024). [CrossRef]
104. Sen, C.; Kashyap, T. Digital Video Watermarking Using DWT for Data Security. *IJARCCE* **2015**, 307–309. [CrossRef]
105. Chan, P.-W.; Lyu, M.R. A DWT-Based Digital Video Watermarking Scheme with Error Correcting Code. In *Proceedings of the Information and Communications Security*; Qing, S., Gollmann, D., Zhou, J., Eds.; Springer: Huhehaote, China, 2003; pp. 202–213.
106. Patil, S.A. Digital Video Watermarking Using Dwt and Pca. *IOSR J. Eng.* **2013**, *3*, 45–49. [CrossRef]
107. Liu, Y.; Zhao, J. A New Video Watermarking Algorithm Based on 1D DFT and Radon Transform. *Signal Process.* **2010**, *90*, 626–639. [CrossRef]
108. Sun, X.-C.; Lu, Z.-M.; Wang, Z.; Liu, Y.-L. A Geometrically Robust Multi-Bit Video Watermarking Algorithm Based on 2-D DFT. *Multimed. Tools Appl.* **2021**, *80*, 13491–13511. [CrossRef]
109. Yang, X.; Zhang, Z.; Jiao, Y.; Li, Z. A Robust Video Watermarking Algorithm Based on Two-Dimensional Discrete Fourier Transform. *Electronics* **2023**, *12*, 3271. [CrossRef]
110. Sang, J.; Liu, Q.; Song, C.-L. Robust Video Watermarking Using a Hybrid DCT-DWT Approach. *J. Electron. Sci. Technol.* **2020**, *18*, 100052. [CrossRef]
111. Mawande, S.; Dakhore, H. Video Watermarking Using DWT-DCT-SVD Algorithms. In Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 18–19 July 2017; pp. 1161–1164.
112. Agarwal, A.; Bhadana, R.; Chavan, S. A Robust Video Watermarking Scheme Using DWT and DCT. *Int. J. Comput. Sci. Inf. Technol. IJCSIT* **2011**, *2*, 1711–1716.
113. Palaiyappan, C.; Raja Jeya Sekhar, T. A Block Based Novel Digital Video Watermarking Scheme Using Dct. *IOSR J. Electron. Commun. Eng.* **2013**, *5*, 34–44. [CrossRef]
114. Zuo, T.; Duan, Y.; Du, Q.; Tao, X. Semantic Security: A Digital Watermark Method for Image Semantic Preservation. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 4645–4649.
115. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 196th ed.; MIT Press: Cambridge, MA, USA, 2016.
116. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
117. Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Le, Q.V.; Mao, M.; Ranzato, M.; Senior, A.; Tucker, P.; et al. Large Scale Distributed Deep Networks. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 3–6 December 2012; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
118. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 3–6 December 2012; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
119. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef]
120. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 2 July 2011; JMLR Workshop and Conference Proceedings. pp. 37–49.
121. Ayinde, B.O.; Zurada, J.M. Deep Learning of Constrained Autoencoders for Enhanced Understanding of Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3969–3979. [CrossRef] [PubMed]
122. Hosseini-Asl, E.; Zurada, J.M.; Nasraoui, O. Deep Learning of Part-Based Representation of Data Using Sparse Autoencoders with Nonnegativity Constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2486–2498. [CrossRef]
123. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
124. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
125. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 6840–6851.
126. Nichol, A.Q.; Dhariwal, P. Improved Denoising Diffusion Probabilistic Models. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8162–8171.
127. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
128. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

https://doi.org/10.3390/s26020444

129. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
130. Ben Jabra, S.; Ben Farah, M. Deep Learning-Based Watermarking Techniques Challenges: A Review of Current and Future Trends. *Circuits Syst. Signal Process.* **2024**, *43*, 4339–4368. [CrossRef]
131. Wang, Z.; Byrnes, O.; Wang, H.; Sun, R.; Ma, C.; Chen, H.; Wu, Q.; Xue, M. Data Hiding with Deep Learning: A Survey Unifying Digital Watermarking and Steganography. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 2985–2999. [CrossRef]
132. Zhong, X.; Das, A.; Alrasheedi, F.; Tanvir, A. A Brief, In-Depth Survey of Deep Learning-Based Image Watermarking. *Appl. Sci.* **2023**, *13*, 11852. [CrossRef]
133. Kandi, H.; Mishra, D.; Gorthi, S.R.K.S. Exploring the Learning Capabilities of Convolutional Neural Networks for Robust Image Watermarking. *Comput. Secur.* **2017**, *65*, 247–268. [CrossRef]
134. Mun, S.-M.; Nam, S.-H.; Jang, H.-U.; Kim, D.; Lee, H.-K. A Robust Blind Watermarking Using Convolutional Neural Network. *Neurocomputing* **2019**, *337*, 191–202. [CrossRef]
135. Das, A.; Zhong, X. A Deep Learning-Based Audio-in-Image Watermarking Scheme. In Proceedings of the 2021 International Conference on Visual Communications and Image Processing (VCIP), Munich, Germany, 5–8 December 2021; pp. 1–5.
136. Zhong, X.; Huang, P.-C.; Mastorakis, S.; Shih, F.Y. An Automated and Robust Image Watermarking Scheme Based on Deep Neural Networks. *arXiv* **2020**, arXiv:2007.02460. [CrossRef]
137. Tavakoli, A.; Honjani, Z.; Sajedi, H. Convolutional Neural Network-Based Image Watermarking Using Discrete Wavelet Transform. *Int. J. Inf. Technol.* **2023**, *15*, 2021–2029. [CrossRef]
138. Ouyang, C.; Wei, Z. Deep Neural Network-Based Image Watermarking in Wavelet Transform Domain. In Proceedings of the 2023 International Conference on Artificial Intelligence and Automation Control (AIAC), Xiamen, China, 17–19 November 2023; pp. 164–167.
139. Lu, J.; Ni, J.; Su, W.; Xie, H. Wavelet-Based CNN for Robust and High-Capacity Image Watermarking. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
140. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015.
141. Tancik, M.; Mildenhall, B.; Ng, R. StegaStamp: Invisible Hyperlinks in Physical Photographs. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2114–2123.
142. Singh, H.K.; Singh, A.K. Digital Image Watermarking Using Deep Learning. *Multimed. Tools Appl.* **2024**, *83*, 2979–2994. [CrossRef]
143. Dasgupta, A.; Zhong, X. Robust Image Watermarking Based on Cross-Attention and Invariant Domain Learning. In Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 13–15 December 2023.
144. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
145. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. HiDDeN: Hiding Data With Deep Networks. *arXiv* **2018**, arXiv:1807.09937.
146. Wen, B.; Aydore, S. ROMark: A Robust Watermarking System Using Adversarial Training. *arXiv* **2019**, arXiv:1910.01221. [CrossRef]
147. Hamamoto, I.; Kawamura, M. Neural Watermarking Method Including an Attack Simulator against Rotation and Compression Attacks. *IEICE Trans. Inf. Syst.* **2020**, *103*, 33–41. [CrossRef]
148. Zhang, L.; Li, W.; Ye, H. A Blind Watermarking System Based on Deep Learning Model. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20–22 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1208–1213.
149. Zhang, H.; Wang, H.; Cao, Y.; Shen, C.; Li, Y. Robust Data Hiding Using Inverse Gradient Attention. *arXiv* **2022**, arXiv:2011.10850. [CrossRef]
150. Hao, K.; Feng, G.; Zhang, X. Robust Image Watermarking Based on Generative Adversarial Network. *China Commun.* **2020**, *17*, 131–140. [CrossRef]
151. Huang, J.; Luo, T.; Li, L.; Yang, G.; Xu, H.; Chang, C.-C. ARWGAN: Attention-Guided Robust Image Watermarking Model Based on GAN. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5018417. [CrossRef]
152. Shedole, S.M.; Santhi, V. Hybrid Deep Learning Based Digital Image Watermarking Using GAN-LSTM and Adaptive Gannet Optimization Techniques. *Multimed. Tools Appl.* **2024**, *84*, 20661–20691. [CrossRef]
153. Karki, B.; Tsai, C.-H.; Huang, P.-C.; Zhong, X. Deep Learning-Based Text-in-Image Watermarking. In Proceedings of the 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 7–9 August 2024.
154. Aberna, P.; Agilandeeswari, L.; Aashich, B. Vision Transformer-Based Watermark Generation for Authentication and Tamper Detection Using Schur Decomposition and Hybrid Transforms. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2023**, *15*, 107–121.

https://doi.org/10.3390/s26020444

155. Liu, Z.; Li, Z.; Zheng, L.; Li, D. Two-Stage Robust Lossless DWI Watermarking Based on Transformer Networks in the Wavelet Domain. *Appl. Sci.* **2023**, *13*, 6886. [CrossRef]
156. Luo, T.; Wu, J.; He, Z.; Xu, H.; Jiang, G.; Chang, C.-C. WFormer: A Transformer-Based Soft Fusion Model for Robust Image Watermarking. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 4179–4196. [CrossRef]
157. Wang, B.; Song, Z.; Wu, Y. Robust Blind Watermarking Framework for Hybrid Networks Combining CNN and Transformer. In Proceedings of the 15th Asian Conference on Machine Learning, Istanbul, Turkey, 27 February 2024; pp. 1417–1432.
158. Chen, W.; Li, Y. RoWSFormer: A Robust Watermarking Framework with Swin Transformer for Enhanced Geometric Attack Resilience. *arXiv* **2024**, arXiv:2409.14829. [CrossRef]
159. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030. [CrossRef]
160. Cui, Y.; Ren, J.; Xu, H.; He, P.; Liu, H.; Sun, L.; Xing, Y.; Tang, J. DiffusionShield: A Watermark for Copyright Protection against Generative Diffusion Models. *arXiv* **2024**, arXiv:2306.04642. [CrossRef]
161. Wen, Y.; Kirchenbauer, J.; Geiping, J.; Goldstein, T. Tree-Ring Watermarks: Fingerprints for Diffusion Images That Are Invisible and Robust. *arXiv* **2023**, arXiv:2305.20030. [CrossRef]
162. Sha, Z.; Li, Z.; Yu, N.; Zhang, Y. DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Generation Models. *arXiv* **2023**, arXiv:2210.06998.
163. Zhang, L.; Liu, X.; Martin, A.V.; Bearfield, C.X.; Brun, Y.; Guan, H. Attack-Resilient Image Watermarking Using Stable Diffusion. *arXiv* **2024**, arXiv:2401.04247.
164. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
165. Tan, Y.; Peng, Y.; Fang, H.; Chen, B.; Xia, S.-T. WaterDiff: Perceptual Image Watermarks Via Diffusion Model. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seul, Republic of Korea, 14–19 April 2024; pp. 3250–3254.
166. Hu, R.; Zhang, J.; Li, Y.; Li, J.; Guo, Q.; Qiu, H.; Zhang, T. SuperMark: Robust and Training-Free Image Watermarking via Diffusion-Based Super-Resolution. *arXiv* **2024**, arXiv:2412.10049.
167. Bui, T.; Agarwal, S.; Collomosse, J. TrustMark: Universal Watermarking for Arbitrary Resolution Images. *arXiv* **2023**, arXiv:2311.18297. [CrossRef]
168. Wang, Y.; Zhu, X.; Ye, G.; Zhang, S.; Wei, X. Achieving Resolution-Agnostic DNN-Based Image Watermarking: A Novel Perspective of Implicit Neural Representation. In Proceedings of the 32nd ACM International Conference on Multimedia, New York, NY, USA, 28 October–1 November 2024; pp. 10354–10362.
169. Kutter, M.; Petitcolas, F.A.P. Fair Benchmark for Image Watermarking Systems. In *Proceedings of the Security and Watermarking of Multimedia Contents*; SPIE: Bellingham, WA, USA, 1999; Volume 3657, pp. 226–239.
170. Duszejko, P.; Piotrowski, Z. 4KSecure: A Universal Method for Active Manipulation Detection in Images of Any Resolution. *Appl. Sci.* **2025**, *15*, 4469. [CrossRef]
171. Numajiri, H.; Hayashi, T. Analysis on Open Data as a Foundation for Data-Driven Research. *Scientometrics* **2024**, *129*, 6315–6332. [CrossRef]
172. Hadhoud, M.M.; Abd El-Samie, F.; El-Khamy, S.E. New Trends in High Resolution Image Processing. In Proceedings of the Fourth Workshop on Photonics and Its Application, Giza, Egypt, 4 May 2004; pp. 2–23.
173. Satone, K.N.; Deshmukh, A.S.; Ulhe, P.B. A Review of Image Compression Techniques. In Proceedings of the 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; Volume 1, pp. 97–101.
174. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. *arXiv* **2021**, arXiv:2102.12092.
175. Dathathri, S.; See, A.; Ghaisas, S.; Huang, P.-S.; McAdam, R.; Welbl, J.; Bachani, V.; Kaskasoli, A.; Stanforth, R.; Matejovicova, T.; et al. Scalable Watermarking for Identifying Large Language Model Outputs. *Nature* **2024**, *634*, 818–823. [CrossRef]
176. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
177. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
178. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
179. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]

https://doi.org/10.3390/s26020444

180. Bas, P.; Filler, T.; Pevný, T. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In Proceedings of the 13th International Conference on Information Hiding, Prague, Czech Republic, 18–20 May 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 59–70.
181. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131.
182. Liu, J.; Liu, D.; Yang, W.; Xia, S.; Zhang, X.; Dai, Y. A Comprehensive Benchmark for Single Image Compression Artifact Reduction. *IEEE Trans. Image Process.* **2020**, *29*, 7845–7860. [CrossRef]
183. Zhang, K.; Li, D.; Luo, W.; Ren, W.; Stenger, B.; Liu, W.; Li, H.; Yang, M.-H. Benchmarking Ultra-High-Definition Image Super-Resolution. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE: Piscataway, NJ, USA; pp. 14749–14758.
184. Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. LAION-5B: An Open Large-Scale Dataset for Training next Generation Image-Text Models. *arXiv* **2022**, arXiv:2210.08402.
185. Bird, J.J.; Lotfi, A. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. *IEEE Access* **2024**, *12*, 15642–15650. [CrossRef]
186. Rahman, M.A.; Paul, B.; Sarker, N.H.; Hakim, Z.I.A.; Fattah, S.A. Artifact: A Large-Scale Dataset with Artificial and Factual Images for Generalizable and Robust Synthetic Image Detection. In Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 8–11 October 2023; pp. 2200–2204.
187. Boychev, D.; Cholakov, R. ImagiNet: A Multi-Content Benchmark for Synthetic Image Detection. *arXiv* **2024**, arXiv:2407.20020.
188. Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; Summers, R.M. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3462–3471.
189. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2217–2226. [CrossRef]
190. Knoche, M.; Hörmann, S.; Rigoll, G. Cross-Quality LFW: A Database for Analyzing Cross-Resolution Image Face Recognition in Unconstrained Environments. In Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, 15–18 December 2021; pp. 1–5.
191. Sood, A.; Mishra, D.; Surya, V.; Singh, H.; Sundaresan, R.; Pal, D.; Dharmaraju, R.; Satish, R.; Mishra, S.; Chavan, N.A.; et al. Challenges and Recommendations for Enhancing Digital Data Protection in Indian Medical Research and Healthcare Sector. *Npj Digit. Med.* **2025**, *8*, 48. [CrossRef] [PubMed]
192. Barni, M.; Bartolini, F. *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications (Signal Processing and Communications)*; CRC: Boca Raton, FL, USA, 2004; ISBN 978-0-8247-4806-7.
193. Pibre, L.; Jérôme, P.; Ienco, D.; Chaumont, M. Deep Learning Is a Good Steganalysis Tool When Embedding Key Is Reused for Different Images, Even If There Is a Cover Source-Mismatch. *arXiv* **2015**, arXiv:1511.04855. [CrossRef]
194. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.-H. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Comput. Surv.* **2023**, *56*, 1–39. [CrossRef]
195. Fuest, M.; Ma, P.; Gui, M.; Schusterbauer, J.; Hu, V.T.; Ommer, B. Diffusion Models and Representation Learning: A Survey. *arXiv* **2024**, arXiv:2407.00783. [CrossRef]
196. Krichen, M.; Abdalzaher, M.S. Performance Enhancement of Artificial Intelligence: A Survey. *J. Netw. Comput. Appl.* **2024**, *232*, 104034. [CrossRef]
197. Zhou, Y.; Chen, S.; Wang, Y.; Huan, W. Review of Research on Lightweight Convolutional Neural Networks. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1713–1720.
198. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
199. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50× Fewer Parameters and <0.5 MB Model Size. *arXiv* **2016**, arXiv:1602.07360.
200. Cheng, H.; Zhang, M.; Shi, J.Q. A Survey on Deep Neural Network Pruning-Taxonomy, Comparison, Analysis, and Recommendations. *arXiv* **2024**, arXiv:2308.06767. [CrossRef] [PubMed]
201. Ma, X.; Lin, S.; Ye, S.; He, Z.; Zhang, L.; Yuan, G.; Tan, S.H.; Li, Z.; Fan, D.; Qian, X.; et al. Non-Structured DNN Weight Pruning—Is It Beneficial in Any Platform? *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4930–4944. [CrossRef]
202. Nagel, M.; Fournarakis, M.; Amjad, R.A.; Bondarenko, Y.; van Baalen, M.; Blankevoort, T. A White Paper on Neural Network Quantization. *arXiv* **2021**, arXiv:2106.08295. [CrossRef]

https://doi.org/10.3390/s26020444

203. Chen, Z.; Tan, X.; Wang, K.; Pan, S.; Mandic, D.; He, L.; Zhao, S. InferGrad: Improving Diffusion Models for Vocoder by Considering Inference in Training. *arXiv* **2022**, arXiv:2202.03751. [CrossRef]
204. Shih, A.; Belkhale, S.; Ermon, S.; Sadigh, D.; Anari, N. Parallel Sampling of Diffusion Models. *arXiv* **2023**, arXiv:2305.16317. [CrossRef]
205. Mahon, S.; Varrette, S.; Plugaru, V.; Pinel, F.; Bouvry, P. Performance Analysis of Distributed and Scalable Deep Learning. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, VIC, Australia, 11–14 May 2020; pp. 760–766.
206. Pham, M.; Li, H.; Yuan, Y.; Mou, C.; Ramachandran, K.; Xu, Z.; Tu, Y. Dynamic Memory Management in Massively Parallel Systems: A Case on GPUs. In *Proceedings of the 36th ACM International Conference on Supercomputing*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 1–13.
207. Sakthidevi, I.; Rajkumar, G.V.; Sunitha, R.; Sangeetha, A.; Krishnan, R.S.; Sundararajan, S. Machine Learning Orchestration in Cloud Environments: Automating the Training and Deployment of Distributed Machine Learning AI Model. In Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 11–13 October 2023; pp. 376–384.
208. Jangda, A.; Pinckney, D.; Brun, Y.; Guha, A. Formal Foundations of Serverless Computing. *Proc. ACM Program. Lang.* **2019**, *3*, 149. [CrossRef]
209. Liang, S.; Jin, S.; Chen, Y. A Review of Edge Computing Technology and Its Applications in Power Systems. *Energies* **2024**, *17*, 3230. [CrossRef]
210. Andriulo, F.C.; Fiore, M.; Mongiello, M.; Traversa, E.; Zizzo, V. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics* **2024**, *11*, 71. [CrossRef]
211. Zeyu, H.; Geming, X.; Zhaohang, W.; Sen, Y. Survey on Edge Computing Security. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Virtual Event, 28–30 June 2020; pp. 96–105.
212. Yan, Z.; Yi, Y.Z.; JiLin, Z.; NaiLiang, Z.; YongJian, R.; Jian, W.; Jun, Y. Federated Learning Model Training Method Based on Data Features Perception Aggregation. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 27–30 September 2021; pp. 1–7.
213. Sáinz-Pardo Díaz, J.; López García, Á. Study of the Performance and Scalability of Federated Learning for Medical Imaging with Intermittent Clients. *Neurocomputing* **2023**, *518*, 142–154. [CrossRef]
214. de Goede, M.; Cox, B.; Decouchant, J. Training Diffusion Models with Federated Learning. *arXiv* **2024**, arXiv:2406.12575. [CrossRef]
215. Qi, Y.; Feng, Y.; Wang, X.; Li, H.; Tian, J. Leveraging Federated Learning and Edge Computing for Recommendation Systems within Cloud Computing Networks. *arXiv* **2024**, arXiv:2403.03165. [CrossRef]
216. Oubadriss, A.; Laassiri, J.; Makrani, A.E. An Overview Comparison between Convolutional Neural Networks and Vision Transformers. In Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security, Ado Ekiti, Nigeria, 26–28 November 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–9.
217. Chen, M.; Mei, S.; Fan, J.; Wang, M. Opportunities and Challenges of Diffusion Models for Generative AI. *Natl. Sci. Rev.* **2024**, *11*, nwae348. [CrossRef] [PubMed]
218. Jamil, S.; Jalil Piran, M.; Kwon, O.-J. A Comprehensive Survey of Transformers for Computer Vision. *Drones* **2023**, *7*, 287. [CrossRef]
219. von Krause, M.; Radev, S.T.; Voss, A.; Quintus, M.; Egloff, B.; Wrzus, C. Stability and Change in Diffusion Model Parameters over Two Years. *J. Intell.* **2021**, *9*, 26. [CrossRef]
220. Ravi, A.; Chaturvedi, V.; Shafique, M. ViT4Mal: Lightweight Vision Transformer for Malware Detection on Edge Devices. *ACM Trans. Embed. Comput. Syst.* **2023**, *22*, 117. [CrossRef]
221. Nag, S.; Liberty, L.; Sivakumar, A.; Yadwadkar, N.J.; John, L.K. Lightweight Vision Transformers for Low Energy Edge Inference. In Proceedings of the Machine Learning for Computer Architecture and Systems 2024, Buenos Aires, Argentina, 30 June 2024.
222. Setyawan, N.; Sun, C.-C.; Hsu, M.-H.; Kuo, W.-K.; Hsieh, J.-W. MicroViT: A Vision Transformer with Low Complexity Self Attention for Edge Device. *arXiv* **2025**, arXiv:2502.05800.
223. Tang, H.; Chen, Y.; Wang, T.; Zhou, Y.; Zhao, L.; Gao, Q.; Du, M.; Tan, T.; Zhang, X.; Tong, T. HTC-Net: A Hybrid CNN-Transformer Framework for Medical Image Segmentation. *Biomed. Signal Process. Control* **2024**, *88*, 105605. [CrossRef]
224. Liu, X.; Liu, J.; Bai, Y.; Gu, J.; Chen, T.; Jia, X.; Cao, X. Watermark Vaccine: Adversarial Attacks to Prevent Watermark Removal. *arXiv* **2022**, arXiv:2207.08178. [CrossRef]
225. Bistroń, M.; Piotrowski, Z. Efficient Video Watermarking Algorithm Based on Convolutional Neural Networks with Entropy-Based Information Mapper. *Entropy* **2023**, *25*, 284. [CrossRef] [PubMed]

https://doi.org/10.3390/s26020444

226. Xu, T.; Mi, P.; Wang, R.; Chen, Y. Towards Faster Training of Diffusion Models: An Inspiration of A Consistency Phenomenon. *arXiv* **2024**. [CrossRef]
227. Li, C.; Zhang, C.; Xu, W.; Xie, J.; Feng, W.; Peng, B.; Xing, W. LatentSync: Audio Conditioned Latent Diffusion Models for Lip Sync. *arXiv* **2024**, arXiv:2412.09262. [CrossRef]
228. Fei, Z.; Fan, M.; Yu, C.; Li, D.; Zhang, Y.; Huang, J. Dimba: Transformer-Mamba Diffusion Models. *arXiv* **2024**, arXiv:2406.01159. [CrossRef]
229. Liu, W.; Zhang, S.Q. HQ-DiT: Efficient Diffusion Transformer with FP4 Hybrid Quantization. *arXiv* **2024**, arXiv:2405.19751.
230. De Falco, F.; Ceschini, A.; Sebastianelli, A.; Le Saux, B.; Panella, M. Quantum Hybrid Diffusion Models for Image Synthesis. *KI–Künstl. Intell.* **2024**, *38*, 311–326. [CrossRef]
231. Uehira, K.; Suzuki, K.; Ikeda, H. Applications of Optoelectronic Watermarking Technique to New Business and Industry Systems Utilizing Flat-Panel Displays and Smart Devices. In Proceedings of the 2014 IEEE Industry Application Society Annual Meeting, Vancouver, BC, Canada, 5–9 October 2014; pp. 1–9.
232. Juefei-Xu, F.; Wang, R.; Huang, Y.; Guo, Q.; Ma, L.; Liu, Y. Countering Malicious DeepFakes: Survey, Battleground, and Horizon. *Int. J. Comput. Vis.* **2022**, *130*, 1678–1734. [CrossRef]
233. Hameleers, M.; van der Meer, T.G.L.A.; Dobber, T. Distorting the Truth versus Blatant Lies: The Effects of Different Degrees of Deception in Domestic and Foreign Political Deepfakes. *Comput. Hum. Behav.* **2024**, *152*, 108096. [CrossRef]
234. Zhao, X.; Gunn, S.; Christ, M.; Fairoze, J.; Fabrega, A.; Carlini, N.; Garg, S.; Hong, S.; Nasr, M.; Tramer, F.; et al. SoK: Watermarking for AI-Generated Content. *arXiv* **2024**, arXiv:2411.18479.
235. SynthID. Available online: https://deepmind.google/models/synthid/ (accessed on 20 November 2025).
236. Agbaje, M.; Awodele, O.; Ogbonna, C. Applications of Digital Watermarking to Cyber Security (Cyber Watermarking). In Proceedings of the Informing Science & IT Education Conference (InSITE), Tampa, FL, USA, 29 June–5 July 2015; pp. 1–11.
237. Dinesh Arokia Raj, A.; Jha, R.R.; Yadav, M.; Sam, D.; Jayanthi, K. Role of Blockchain and Watermarking Toward Cybersecurity. In *Multimedia Watermarking: Latest Developments and Trends*; Kumar Sahu, A., Ed.; Springer Nature: Singapore, 2024; pp. 103–123. ISBN 978-981-99-9803-6.
238. Blake, S. Embedded Blockchains: A Synthesis of Blockchains, Spread Spectrum Watermarking, Perceptual Hashing & Digital Signatures. *arXiv* **2020**, arXiv:2009.00951.
239. Singh, B.; Sharma, M.K. Efficient Watermarking Technique for Protection and Authentication of Document Images. *Multimed. Tools Appl.* **2022**, *81*, 22985–23005. [CrossRef]
240. Kapre, B.S.; Rajurkar, A.M.; Guru, D.S. The Blind Robust Video Watermarking Scheme in Video Surveillance Context. *Multimed. Tools Appl.* **2024**, *83*, 38999–39025. [CrossRef]
241. Wazirali, R.; Ahmad, R.; Al-Amayreh, A.; Al-Madi, M.; Khalifeh, A. Secure Watermarking Schemes and Their Approaches in the IoT Technology: An Overview. *Electronics* **2021**, *10*, 1744. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Sensors Editorial Office
E-mail: sensors@mdpi.com
www.mdpi.com/journal/sensors

Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-6727-1