sensors

Special Issue Reprint

Deep Learning Technology and Image Sensing

Edited by
Sukho Lee and Dae-Ki Kang

mdpi.com/journal/sensors

MDPI

Deep Learning Technology and Image Sensing

Deep Learning Technology and Image Sensing

Guest Editors

Sukho Lee
Dae-Ki Kang

MDPI

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

Guest Editors

Sukho Lee	Dae-Ki Kang
Dongseo University	Dongseo University
Busan	Busan
Republic of Korea	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/Deep_Learning_Technology_Image.

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 978-3-7258-2813-5 (Hbk)
ISBN 978-3-7258-2814-2 (PDF)
https://doi.org/10.3390/books978-3-7258-2814-2

Contents

sensors

MDPI

Editorial

Deep Learning Technology and Image Sensing

Suk-Ho Lee * and Dae-Ki Kang

Department Computer Engineering, Dongseo University, Busan 47011, Republic of Korea; dkkang@dongseo.ac.kr
* Correspondence: petra@g.dongseo.ac.kr

The scientific landscape is constantly evolving, marked by groundbreaking advancements in imaging, sensing, and machine learning that expand the realms of possibility across various disciplines. Today, deep learning-based computing technology plays a pivotal role in enhancing the quality and reliability of image recognition data. For instance, in the realm of autonomous driving, deep learning-based fusion of data from front camera sensors and radars enables the significant improvement of sensor performance. Additionally, deep learning-driven computer vision technologies contribute to enhancing smartphone camera applications, enabling functionalities such as face recognition, panorama photography, depth/geometry detection, and high-quality magnification and detection. This Special Issue, entitled "Deep Learning Technology and Image Sensing", encompasses all topics related to applications utilizing deep learning-based image and video sensing technologies. In this editorial overview, we present eleven papers that are published in this Special Issue that demonstrate innovative approaches and methodologies spanning diverse domains. We divide the domains into applications across medical imaging and healthcare, image enhancement, object detection, and innovation in image sensing technologies.

Several papers in the domain of medical imaging and healthcare applications focus on medical imaging and its applications in healthcare. From liver segmentation and brain tumor classification to the automated detection of optic cup and disc edges in glaucoma patients, these studies demonstrate the power of deep learning frameworks in improving the diagnostic accuracy and clinical decision-making.

Liver segmentation in MRI images poses challenges due to the variable characteristics and lack of HoU-based preprocessing. To tackle this problem, in [1], the authors explore state-of-the-art segmentation networks on T1-weighted MRI scans from a public dataset. A novel cascaded network is proposed, achieving superior results with a DSC of 95.15% and IoU of 92.10% on expert-labeled liver masks. The framework demonstrates high accuracy and holds promise for medical imaging applications.

Rapid and accurate brain tumor detection is crucial for patient health. Despite recent advancements in AI methods, precise diagnoses remain challenging. The work in [2], entitled "A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features", proposes a novel approach using an ensemble of hand-crafted features and deep features from VGG16. The ensemble improves discrimination, achieving a 99% accuracy when classified using SVM or KNN. This method demonstrates reliability for MRI-based tumor detection, offering robustness and potential real-world deployment. Validation through cross-tabulated data confirms the model's good performance.

The study in [3], entitled "Identifying the Edges of the Optic Cup and the Optic Disc in Glaucoma Patients by Segmentation", focuses on automating the detection of optic cup and optic disc edges in fundus images of glaucoma patients, crucial for early diagnosis. Utilizing a modified U-Net model, we evaluate the segmentation performance across multiple datasets. The postprocessing techniques utilized enhance visualization for improved cup-to-disc ratio analysis. The results demonstrate a promising segmentation efficiency, which is particularly beneficial for clinical applications.

Advancements in image enhancement techniques are evident in the papers exploring image-to-image translation in astronomy, super-resolution imaging, and depth map super-resolution. These studies leverage sophisticated algorithms to enhance the image quality,

Citation: Lee, S.-H.; Kang, D.-K. Deep Learning Technology and Image Sensing. *Sensors* **2024**, *24*, 5130. https://doi.org/10.3390/s24165130

Received: 28 July 2024
Accepted: 31 July 2024
Published: 8 August 2024

extract meaningful information, and address challenges in various imaging modalities.

In the work of [4], entitled "Hubble Meets Webb, Image-to-Image Translation in Astronomy", the authors explore the translation of Hubble Space Telescope (HST) data into James Webb Space Telescope (JWST) imagery using various techniques. The proposed method emphasizes the importance of image registration and introduces uncertainty estimation to enhance the translation reliability. This approach aids in preparatory strategies for JWST observations, offering predictive insights when JWST data are unavailable, making it the first attempt at sensor-to-sensor image translation in astronomy.

Meanwhile, the study in [5], entitled "Kernel Estimation Using Total Variation Guided GAN for Image Super-Resolution", addresses artifacts in image super-resolution. The authors propose a Total Variation Guided KernelGAN focusing on structural details. Their experimental results demonstrate this method's effectiveness in accurately estimating kernels, leading to improved super-resolution algorithm performance.

Another approach for super-resolution is descibed in the study entitled "Fully Cross-Attention Transformer for Guided Depth Super-Resolution" [6], which presents a fully transformer-based network for depth map super-resolution, addressing issues in existing guided super-resolution methods. This approach utilizes a cascaded transformer module with a novel cross-attention mechanism, seamlessly guiding the color image into the depth upsampling process. Leveraging a window partitioning scheme ensures linear complexity for high-resolution images and extensive experiments demonstrate the superiority of the proposed method over state-of-the-art approaches.

Besides super-resolution, a low-light image enhancement technique has also been proposed in the work in [7], entitled "Low-Light Image Enhancement Using Hybrid Deep-Learning and Mixed-Norm Loss Functions". The authors propose a method for enhancing low-light images using a hybrid deep learning network and mixed-norm loss functions. Their approach includes a decomposition-net to separate reflectance and illuminance, an illuminance enhance-net to improve illuminance while reducing artifacts, and a chroma-net to mitigate color distortion. YCbCr channels are utilized for training and restoration to account for RGB channel correlations. Mixed-norm loss functions enhance stability and reduce blurring by reflecting reflectance, illuminance, and chroma properties. The experimental results show significant subjective and objective improvements compared to state-of-the-art deep-learning methods.

The integration of artificial intelligence and object detection plays a pivotal role in driving innovation across different fields. The papers in this category present real-time action sensing and detection systems, weakly supervised object detection methods, and novel approaches for brain–computer interfaces. These studies underscore the importance of leveraging AI and ML techniques to automate tasks, improve efficiency, and enable the development of new applications.

The work in [8] ("A Real-Time Subway Driver Action Sensing and Detection Based on Lightweight ShuffleNetV2 Network") proposes a lightweight two-stage model for the real-time monitoring of subway drivers' actions using surveillance cameras. Ensuring subway train safety relies heavily on the actions of drivers. To this end, the model utilizes MobileNetV2-SSDLite for driver detection and an enhanced ShuffleNetV2 network for action recognition, achieving a superior performance over existing models and meeting runtime requirements for practical deployment.

Meanwhile, the authors of [9] ("Instance-Level Contrastive Learning for Weakly Supervised Object Detection") propose instance-level contrastive learning (ICL) in order to mine reliable instance representations from images via contrastive loss.They introduce instance-diverse memory updating (IMU) to capture diverse instances and memory-aware instance mining (MIM) to enhance object instance retrieval. Additionally, memory-aware proposal sampling (MPS) is utilized to balance positive–negative sample learning. The experimental results demonstrate significant gains in the detection accuracy compared to the baselines, showcasing the effectiveness of the approach.

The final set of papers in this Special Issue explores groundbreaking innovations in sensing and imaging technologies. From non-line-of-sight imaging using echolocation to quick-response eigenface analysis schemes for brain–computer interfaces, these studies push the boundaries of what is possible in remote sensing, imaging, and neurotechnology. By introducing novel methodologies and leveraging cutting-edge technologies, these papers pave the way for future advancements in sensing and imaging applications.

The work in [10], entitled "Deep Non-Line-of-Sight Imaging Using Echolocation", introduces a novel approach to non-line-of-sight (NLOS) imaging using acoustic equipment inspired by echolocation. This paper offers a promising alternative to optical NLOS imaging. Unlike optical systems, which rely on diffused light, our method leverages echoes to visualize hidden scenes. Traditional acoustic NLOS methods suffer from noise and long acquisition times. To address this, the authors propose simultaneous echo collection and deep learning models to overcome interference challenges. The model successfully reconstructs hidden object outlines, offering a promising alternative to optical NLOS imaging.

The work in [11], entitled "A Novel Quick-Response Eigenface Analysis Scheme for Brain–Computer Interfaces", introduces a novel quick-response eigenface analysis (QR-EFA) scheme for motor imagery in brain–computer interfaces (BCIs). Leveraging EEG signals in a standardized QR image domain, they combine EFA with a convolutional neural network (CNN) for neuro image classification. To address non-stationary BCI data and non-ergodic characteristics, they employ effective neuro data augmentation during training. QR-EFA enhances the classification accuracy by maximizing similarities in domain, trial, and subject directions. QR-EFA enhances the classification accuracy by maximizing similarities in domain, trial, and subject directions. The experimental results on BCI competition datasets demonstrate significant performance improvement over previous methods, achieving classification accuracies of 97.87%.

Lastly, the authors of [12] aim to utilize technology to develop systems capable of recognizing Arabic Sign Language (ArSL) through deep learning techniques. They propose a hybrid model designed to capture the spatio-temporal aspects of sign language, including both letters and words. This hybrid model incorporates a convolutional neural network (CNN) to extract spatial features from sign language data and a Long Short-Term Memory (LSTM) network to capture spatial and temporal characteristics for handling sequential data, such as hand movements.

In conclusion, the diverse range of papers presented in this editorial overview highlights the interdisciplinary nature of research in deep learning-based imaging and sensing. These studies not only showcase the latest advancements in this area of research but also provide valuable insights and methodologies that have the potential to impact various fields, from healthcare and astronomy to robotics and beyond.

Acknowledgments: This research was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2023RIS-007).

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

References

1. Hossain, M.S.A.; Gul, S.; Chowdhury, M.E.H.; Khan, M.S.; Sumon, M.S.I.; Bhuiyan, E.H.; Khandakar, A.; Hossain, M.; Sadique, A.; Al-Hashimi, I.; et al. Deep Learning Framework for Liver Segmentation from T1-Weighted MRI Images. *Sensors* **2023**, *23*, 8890. [CrossRef] [PubMed]
2. Kibriya, H.; Amin, R.; Kim, J.; Nawaz, M.; Gantassi, R. A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features. *Sensors* **2023**, *23*, 4693. [CrossRef] [PubMed]
3. Tadisetty, S.; Chodavarapu, R.; Jin, R.; Clements, R.J.; Yu, M. Identifying the Edges of the Optic Cup and the Optic Disc in Glaucoma Patients by Segmentation. *Sensors* **2023**, *23*, 4668. [CrossRef] [PubMed]
4. Kinakh, V.; Belousov, Y.; Quétant, G.; Drozdova, M.; Holotyak, T.; Schaerer, D.; Voloshynovskiy, S. Hubble Meets Webb: Image-to-Image Translation in Astronomy. *Sensors* **2024**, *24*, 1151. [CrossRef]
5. Park, J.; Kim, H.; Kang, M.G. Kernel Estimation Using Total Variation Guided GAN for Image Super-Resolution. *Sensors* **2023**, *23*, 3734. [CrossRef]

6. Ariav, I.; Cohen, I. Fully Cross-Attention Transformer for Guided Depth Super-Resolution. *Sensors* **2023**, *23*, 2723. [CrossRef] [PubMed]
7. Oh, J.; Hong, M.C. Low-Light Image Enhancement Using Hybrid Deep-Learning and Mixed-Norm Loss Functions. *Sensors* **2022**, *22*, 6904. [CrossRef] [PubMed]
8. Shen, X.; Wei, X. A Real-Time Subway Driver Action Sensoring and Detection Based on Lightweight ShuffleNetV2 Network. *Sensors* **2023**, *23*, 9503. [CrossRef] [PubMed]
9. Zhang, M.; Zeng, B. Instance-Level Contrastive Learning for Weakly Supervised Object Detection. *Sensors* **2022**, *22*, 7525. [CrossRef] [PubMed]
10. Jang, S.; Shin, U.H.; Kim, K. Deep Non-Line-of-Sight Imaging Using Echolocation. *Sensors* **2022**, *22*, 8477. [CrossRef] [PubMed]
11. Choi, H.; Park, J.; Yang, Y.M. A Novel Quick-Response Eigenface Analysis Scheme for Brain–Computer Interfaces. *Sensors* **2022**, *22*, 5860. [CrossRef] [PubMed]
12. Noor, T.H.; Noor, A.; Alharbi, A.F.; Faisal, A.; Alrashidi, R.; Alsaedi, A.S.; Alharbi, G.; Alsanoosy, T.; Alsaeedi, A. Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model. *Sensors* **2024**, *24*, 3683. [CrossRef] [PubMed]

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

A Novel Quick-Response Eigenface Analysis Scheme for Brain–Computer Interfaces

Hojong Choi [1], Junghun Park [2] and Yeon-Mo Yang [2,]*

[1] Department of Electronic Engineering, Gachon University, 1342 Seongnam-daero, Seongnam 13306, Korea
[2] School of Electronic Engineering, Kumoh National Institute of Technology, 61 Daehak-ro, Gumi 39177, Korea
* Correspondence: yangym@vivaldi.kumoh.ac.kr; Tel.: +82-54-478-7488

Abstract: The brain–computer interface (BCI) is used to understand brain activities and external bodies with the help of the motor imagery (MI). As of today, the classification results for EEG 4 class BCI competition dataset have been improved to provide better classification accuracy of the brain computer interface systems (BCIs). Based on this observation, a novel quick-response eigenface analysis (QR-EFA) scheme for motor imagery is proposed to improve the classification accuracy for BCIs. Thus, we considered BCI signals in standardized and sharable quick response (QR) image domain; then, we systematically combined EFA and a convolution neural network (CNN) to classify the neuro images. To overcome a non-stationary BCI dataset available and non-ergodic characteristics, we utilized an effective neuro data augmentation in the training phase. For the ultimate improvements in classification performance, QR-EFA maximizes the similarities existing in the domain-, trial-, and subject-wise directions. To validate and verify the proposed scheme, we performed an experiment on the BCI dataset. Specifically, the scheme is intended to provide a higher classification output in classification accuracy performance for the BCI competition 4 dataset 2a (C4D2a_4C) and BCI competition 3 dataset 3a (C3D3a_4C). The experimental results confirm that the newly proposed QR-EFA method outperforms the previous the published results, specifically from 85.4% to 97.87% ± 0.75 for C4D2a_4C and 88.21% ± 6.02 for C3D3a_4C. Therefore, the proposed QR-EFA could be a highly reliable and constructive framework for one of the MI classification solutions for BCI applications.

Keywords: motor imagery classification; eigenface analysis; quick response neuro images; image data augmentation; standardized and sharable quick response eigenfaces

Citation: Choi, H.; Park, J.; Yang, Y.-M. A Novel Quick-Response Eigenface Analysis Scheme for Brain–Computer Interfaces. *Sensors* **2022**, *22*, 5860. https://doi.org/10.3390/s22155860

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 1 July 2022
Accepted: 1 August 2022
Published: 5 August 2022

1. Introduction

The brain–computer interface (BCI) is a continuously developing technology that implement the brain's thought to manipulate external bodies without thinking through the human nerves to the hands or feet [1,2]. Thus, BCI systems are intended to generate new communication channels for other human parts [3,4]. There are two types of justifications available. One is uni-directional communication from a brain to a computer and the other is bi-direction communication between a brain and computer, as well as brain-to-brain. The current state of art technology for the BCI is bi-direction BCI technology or brain to brain interface (B2BI) [5]. The B2BI functions via on one hand as a brain–computer interface (BCI) to retrieve messages and, on the other hand, a computer-brain interface (CBI) [6]. The communication paths that do not go through human's nerves could help people with physical disabilities by controlling objects, such as a wireless mouse [7]. The signals that are generated from the human brain are used to implement the systems, and they are extremely complex; such signals are recorded in electronic system, i.e., an electroencephalogram (EEG) [8]. In the beginning of EEG research, EEG measurement used invasive and painful methods, in which several electrodes were inserted directly into the human skin [9]. The non-invasive methods of attaching the electrodes to the human scalps have been used because they are non-painful, user-friendly, and economical ways to obtain EEG signals [10].

In particular, the classification of motor imagery (MI) is researched because EEG signals are shown when the right and left hands are related, respectively [11].

In the BCI systems, raw EEG signals need to be processed with filtering, extraction, and classification techniques [12]. We describe the whole BCI workflow based on the OSI 7-layer model; the OSI 7-layer model is a kind of the communication model. Therefore, it could be useful for readers to understand the feature extraction and classification algorithms in the signal processing workflow. In OSI 7-layer model, there are two types of disciplines. One is a datagram for a layered network hierarchy, and the other is a conceptual model for multistage processing. In the layered hierarchy model, there are the physical, medium access control, network, transport, or routing, etc. In the conceptual model for multistage processing, the focus is on multistage processing, such as the input, data acquisition, preprocessing, decision/detection, and hypothesis test. In this article, based on the previous literature of IEEE [11], we compared the similarity between the BCI dataflow and OSI 7-layer in the conceptual model for better understanding of the signal progressing in BCIs. According to N. Khadijah et al., in the previous related work [13], we used the open system interconnection (OSI) layer, as shown in Figure 1. In the BCI model, we used a layer architecture staring from data acquisition (layer 1), going through pre-processing (layer 2), approaching at feature extraction (layer 3), and then ending at classification (layer 4).

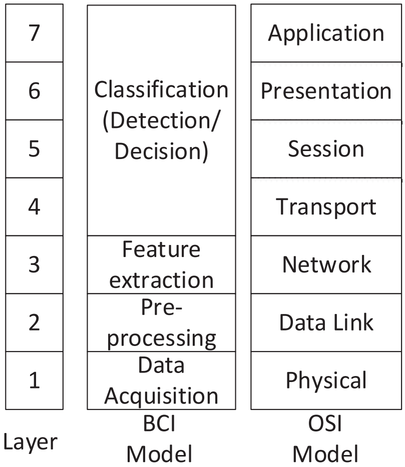


Figure 1. The BCI layer model compared to OSI network model.

In the signal processing steps, feature extraction was used; generally, a filter block is applied before feature extraction [14–16]. There are several feature extraction methods, such as wavelet transform, short-time Fourier transform (STFT), common spatial pattern (CSP), regularized CSP (RCSP), common spatio-spectral pattern (CSSP), common sparse spectral spatial pattern (CSSSP), etc. [8]. As a representative visualization method, short time Fourier transform (STFT), spectrogram, wavelet, etc., were used [17]. Among those methods, the CSP is one of the most widely accepted feature extraction methods [18]. However, the CSP method has overfitting problems when data are sparse in the domain. The CSP method is also noise sensitive, so it is difficult to classify multi-class EEG signals. Blankertz et al. focused how a CSP filter affected the EEG signals [19]. The combination of CSP has been used by many researchers, including Lotte and Guan [20]. Their idea was used to regularize CSP and substitute the normal CSP. The limitation is that the researchers were required to use the full set of data for the training and testing sections. The training data should have a larger number of data than the test data. The full set of data that were used, especially the test data, may lead to an overestimation in performance because all the information is being used. To classify the MI for a subject, the algorithms required the use of other subjects, as well. Therefore, the algorithms are not suitable for classifying a single subject dataset. The extended algorithm of the CSP was used by Lemm et al. [10]. This algorithm is known as CSSP. The CSSP method introduced a delay to the system, so that the spectral filter was included in the system. The simulation results, except for six datasets, showed better accuracy than CSP.

In current EEG classification studies, linear discriminant analysis (LDA), support vector machine (SVM), deep learning, etc., were used [21]. Deep learning (DL) is a machine learning (ML) method used in various fields, such as the voice and video processing fields [22]. The DL works successfully on non-linear and non-stationary data, and it works efficiently even in the fields that are difficult for humans to distinguish [23]. The convolutional neural network (CNN) is one of the DL algorithms that is widely used for data classification [24]. Due to CNN characteristics, there are attempts to apply CNN for EEG signal classification. To use the CNN method specialized for image classification, various methods to visualize EEG signal are being studied [25]. Visualization of these EEG signals was used to help improve the classification performance of DL models using CNN algorithms. Compared to those traditional classification algorithms, such as LDA and SVM, the deep learning model requires the use of large numbers of the dataset [25]. Therefore, the limited numbers of the dataset need to be increased using a data augmentation technique. Considering the finite training data in BCIs, Huang et at.al proposed a data augmentation scheme via a CNN [26]. The data augmentation has accomplished by mixing and recombining the images. Lee et al. convers the lingering problem of zero-training by utilizing a proposed CNN model connected to P300 information [27].

In the EFA method, the calculated eigenface coefficient was used as a feature for data classification. In this case, it is possible to reconstruct the pictures using the eigenface coefficients. We proposed a novel EEG signals classification method, called quick response EFA (QR-EFA) utilizing deep knowledge in standardized and sharable QR image formulations. The QR-EFA is subjected in the EEG signal preprocessing stage. Image reconstruction or data augmentation using QR-EFA has been shown to generate QR image features suitable for CNN algorithms. To overcome the constraints, such as the trial numbers and limited BCI competition dataset in EEG, we proposed an innovative classification technique that was organically designed for EFA and CNN based on data augmentation. After learning the impulse response filter for the BCI competition IV dataset 2a (C4D2a_4c) [28] and BCI competition III dataset 3a (D3D3a_4c) [29], using a seven-layer simple CNN model (one input and output, one convolution, one pooling, and three fully connected), the test EEG data classification performance was measured according to the transfer function. As of today, the highest classification results for EEG 4 classes on BCI competition IV 2a is 85.4% [1]. Based on this observation, our proposed QR-EFA method could provide higher accuracy performance of MI classification for competition dataset in BCIs.

2. Materials and Principles

The proposed EFA algorithm is a feature extraction method from EEG data that builds up neuro images, emphasizing the discriminability of classes; the feature is a kind of tool.

The QR-EFA method was formulated based on the previous EFA result. From egienfaces derived from the previous EFA, we will provide the description. First, the eigenface is asymmetric in horizontal and vertical directions, and the image size is excessively bigger than the size that CNN can process. Second, in BCI competition dataset, there are many discrepancies in data sizes among number of classes, trials, and subjects. Based on this fact, we proposed QR eigenface to provide a symmetrical, standardized, and sharable sizes in images. In the framework level, QR-EFA consists of the parts, such as EFA, eigenface restructuring, data augmentation, and CNN. Figure 2 shows the relationship between the previous proposed EFA and proposed QR-EFA.

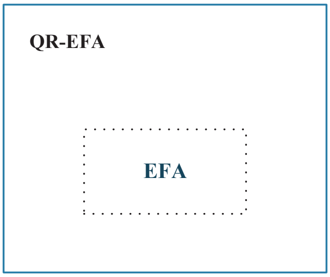


Figure 2. Overall collaboration between QR-EFA and EFA.

The EFA is different PCA image recognition type for dimension reduction [30]. The fundamental EFA method is depicted in Figure 3. The EEG data were preprocessed. As shown in steps 1–3, EEG data were converted to image data to build up eigenface. From eigenface, coefficients, which we called features, were extracted for training and testing procedures. Afterwards, the classes were classified for the next step.

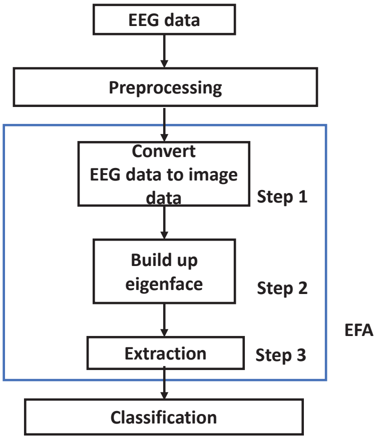


Figure 3. The EFA algorithm procedure.

Let us define the next step of the data interpretation after EFA procedure. We also used the same data interpretation techniques for classification.

Step 1: The three-dimensional (3D) converted EEG image data could be separated as time, channels, and trials. The data were recognized in each separated 3D converted EEG image data, and the generated data could be differentiated, depending on the viewpoint for each 3D direction.

Step 2: From the differentiated 3D EEG image data, the covariance matrix must be obtained. Afterwards, we could determine and build up the eigenfaces.

Step 3: The training data were projected to obtain the requested features. The dataset was composed of, or represented with, '*time (S)*', '*channels (C)*', and '*trials (N)*', as described in Equation (1).

$$Dataset = time\ (S) \times channels(C) \times trials\ (N) \tag{1}$$

EFA is a method to obtain the differentiated EEG data with different directions. The EEG image data with the time, channels, and trials were combined into the dataset, which is an infinite number (N) of the image data with the same manner as that illustrated in Equation (2). In other words, we consider 3D data in two-dimensional (2D) images by combining or concatenating 'time' and 'channel' data together as in $T = SC$ (time × channel). In fact, the derived tentative dataset, T is composed of 'time' and 'channel' components in series. The desired dataset or image matrix was obtained by rearranging the component T in horizontal direction and component N in vertical direction as shown in Figure 4. Consequently, Equation (2) shows the final dataset or 2D images.

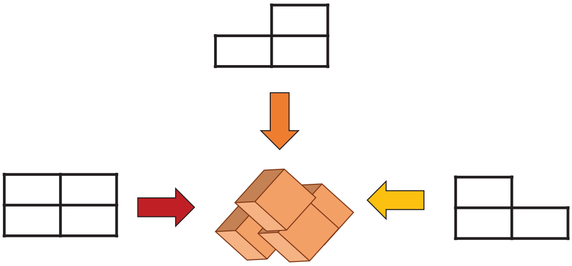


Figure 4. Data analysis on the viewpoint direction.

$$Dataset = TN \ where \ T = SC \tag{2}$$

The eigenface was then obtained from differentiated 3D EEG image, and newly obtained image data Φ and value Ψ need to be calculated.

$$\Phi_k = Dataset_k - \Psi_k, k = 1, 2, \ldots, N \tag{3}$$

The covariance matrix extracted from the differentiated 3D EEG image data without the mean value is obtained in Equation (4).

$$C = \frac{1}{L} \sum_{k=1}^{L} \Phi_k \Phi_k^T \tag{4}$$

Let us define the eigenvectors of X with eigenvalues of L of the covariance matrix C after solving the following equation $CX = kX$. Among the vectors extracted from this matrix, the k vectors were chosen. The eigenfaces must be extracted with only training eigenface Γ_{train}. Subsequently, the training features were extracted from the training eigenface and data. The extracted eigenface coefficients were projected in Equation (5).

$$\Omega_{train} = \Phi_{train} \, \Gamma_{train} \tag{5}$$

The weight coefficient Ω_{train} was obtained to be a training feature. Equation (6) shows how to obtain the feature coefficient Ω_{test}. The eigenspace was trained, and the EEG test data can be classified as shown in Equation (6).

$$\Omega_{test} = \Gamma_{train} \, \Phi_{test} \tag{6}$$

3. QR-EFA

3.1. Idea Formulation for QR-EFA

The description of the procedure and flowchart for QR-EFA is shown. The proposed QR-EFA procedure is given below.

1. Perform the process with channel direction.
2. Original image eigenface formulation.
3. Confine the number of eigenfaces, up to the number of classes required and trials in common among subjects.
4. Adjust eigenface size horizontally with number of classes and vertically with number of trials.
5. Modify image eigenface formulation by multiplying a brightness factor with steps 3 and 4.

6. Obtaining training coefficients by projecting train images to the eigenface.
7. Implement QR neuro train images with the coefficients using the result of step 6.
8. Obtaining testing coefficients by projecting test images to the eigenface.
9. Implement QR neuro test images with the coefficients using the result of step 8.
10. Neuro image augmentation to diversify the limited neuro train images.
11. CNN training process with the results produced, using the result of step 10.

Figure 5 shows the flowchart of the procedure for QR-EFA, described above.

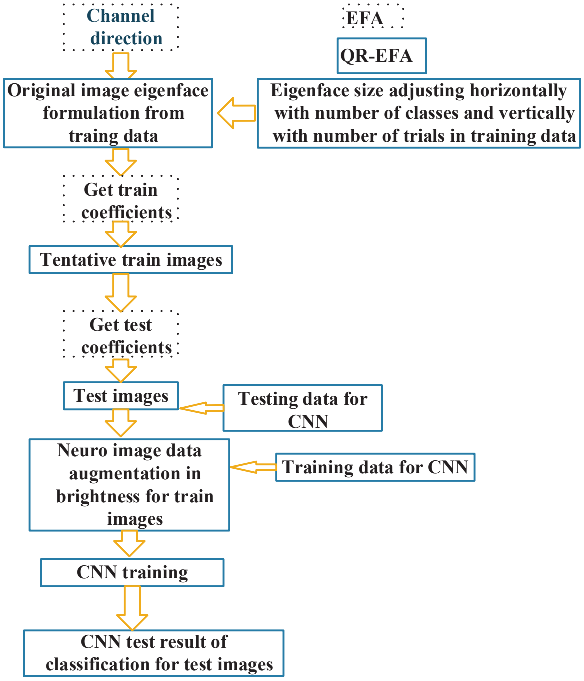

Channel direction
EFA
QR-EFA
Original image eigenface formulation from traing data
Eigenface size adjusting horizontally with number of classes and vertically with number of trials in training data
Get train coefficients
Tentative train images
Get test coefficients
Test images
Testing data for CNN
Neuro image data augmentation in brightness for train images
Training data for CNN
CNN training
CNN test result of classification for test images

Figure 5. Overall flowchart for QR-EFA (solid line) including EFA (dashed line). Cross reference in the above data flowchart for step 1~11.

3.2. Data Augmentation for the Limited Training Data

After we finished the EFA method for the C4D2a_4C and C3D3a_4C datasets, the features were obtained. Based on the features, QR code-like brain neuro images will be formulated. There are two different image types. One is for the training process, and the other is for the testing process. The amount of training data set is definitely limited in BCI competitions, so we need a data augmentation process to multiply the limited images to the desired number of training images. To augment the limited training images, we adapted a brightness control for the input training images, according to the following probability model. For the given normal probability density distribution (PDF), with mean (μ) and standard deviation (σ) x ~ N (μ, σ^2), the statistical random data *x[n]* will be obtained as follows in Equation (7) [31].

$$\begin{gathered}\text{PDF}: f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}, -\infty < x < \infty \\ x = \mu + \sigma\end{gathered} \tag{7}$$

The MATLAB code for training image data augmentation by Gaussian blur (brightness and contrast implication or image bacteria culturing per pixel) is as follows (Box 1).

Box 1. The MATLAB code for training image data augmentation by Gaussian brightness control.

```
rand_img = 1 + (0.7).* randn(is1,is2);
% 1/0.7 (example of is1,is2,is3: 18,16,30)
 timg = trimages(:,:,j).* rand_img;
% neuro image culture by multiplication
```

The data augmentation per pixel was demonstrated to enhance the performance of deep learning approaches by reducing overfitting problems. The overfitting or high variance in ML models are produced if the dataset used to "teach" the model is greater than the testing accuracy. The overfitting problems can be generated when the model has very high errors in the testing dataset. According to the data science literature, CNN needs a huge number of neuro images for the model to be trained effectively [32]. This proposed technique could be helpful for increasing the performance of the model, thereby reducing overfitting problems from the EEG data. Most BCI competition datasets for classification and object detection datasets have a few hundred to thousands of neuro images. Considering the rule of larger numbers or the invariance property of CNN, the objects can even be classified when they are visible in different sizes, orientations, or potentials on channels. Hence, we can take the limited neuro competition dataset of images and transform the objects into different sizes by controlling for brightness, scaling in size, rotating in orientation, or zooming in and out. Through this type data augmentation, we can create intense and diverse neuro image datasets with the extent of variations.

3.3. Covolution Neural Network for QR-EFA

Compared to conventional classifiers, such as LDA, SVM, and multilayer perceptron (MLP), the convolution neural network (CNN) is the most widely used multi-layered neural network ML method for image classification [19]. A general CNN method learns an input image through convolution, pooling, and fully connected layers, so it acts as a classifier [33]. The convolutional layer is composed of feature maps, with several different weight vectors; each feature map is calculated from the input image. The CNN-based image classification models, including ResNet and DarkNet, are widely known [33]. Using an appropriate ML model and neural network structure, according to the target images to be classified, is very important for obtaining high classification accuracy [34]. To better show that EEG images are easily classified using the QR-EFA method, a simple CNN model using only one convolutional layer, a pooling layer, and a fully connected layer was used.

In Figure 6, a simple one convolution and mean pooling are shown to reduce a computation time and minimize the task burden. The size of input images is 12×10 and then it goes through a convolution layer with kernel or filter size of 7×1. After the layer process is done, it would be 6×10 output; then, it goes through 2×2 mean and max pooling layers. After processing a fully connected layer, we finally obtained the output with the class classification output for BCI dataset.

Figure 6 shows a CNN model that classifies the EEG images converted by the QR-EFA method. The convolution layer of the CNN model, which consists of sixteen filters of size 9×3 and a pooling layer of size 2×2, was used. Using softmax and cross entropy methods, two class classification results are obtained. As hyper parameters, the learning rate was set to 0.0001, and a batch size of 512 and momentum optimizer were used. The CNN model was trained for sufficient learning, and training was set to finish early to prevent overfitting problems.

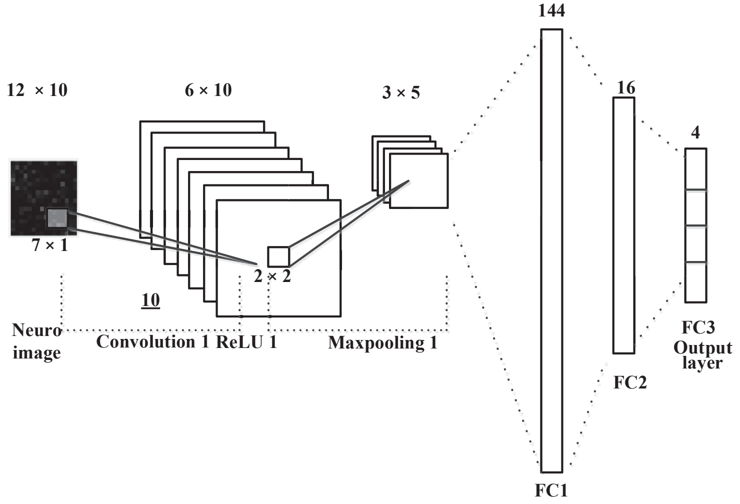


Figure 6. The CNN for neuro images in QR-EFA.

4. Experimental Results

The QR codes are widely used for quick response as matrix barcodes. Just as with QR codes, we also obtained the images, so we called them QR images. After performing QR image data, we need further steps. In obtaining QR images for training and testing images, there are three factors to consider for better classification, in order to have accurate, robust, and reliable results. First, let us define some terminologies, such as the domain-wise similarity among domains, trial-wise similarity among trials, and subject-wise similarity among subjects. Domain-wise similarity is the degree of similarity that indicates some common characteristics on the eigenfaces between the training and testing domains. Trial- and subject-wise similarity were degrees of similarity among trials and subjects, respectively. We could recognize the similarity from QR images. Our proposed QR-EFA algorithm maximizes the similarities, with the respect to domains, trials, and subjects. As result, it could generate unique features per domain, so it is possible to discriminate the classes efficiently.

The background regarding EEG datasets from BCI competitions needs to be explained. To validate the proposed method, we used two EEG datasets, such as the BCI competition III dataset 3a (C3D3a_4C) and BCI competition IV dataset 2a (C4D2a_4C), for four classes. Specifically, C3D3a_4C is dataset from three subjects or participants, and C4D2a_4C is dataset from nine subjects, which are the off-line, publicly available, and open accessible dataset of the BCI competition database. Hence, this article focuses on the C3D3a_4c and C4D2a_4C. Table 1 shows the detailed number of trials per subject for the C3D4a_4C used in this article.

Table 1. The C3D3a_4C dataset composed of three subjects and predefined number of experimental trials.

Subject	Class (# of Trials)			
	Left (L)	Right (R)	Foot (F)	Tongue (T)
1	45	45	45	45
2	30	30	30	30
3	30	30	30	30

The detailed information of the property for C3D3a_4C is given as follows (Box 2).

Box 2. The detailed information of the property for C3D3a_4C.

```
comment1: 'dataset: C3D3a_4C'
date: '2021.12.28'
madeby: '4C'
affiliation: 'KNIT'
window: 'offset: 3.5, length: 2'
subject: 'subject #: 1'
prefiltering: 'off'
s: 250 (sample/sec)
c: [1 × 60 cell]
x: [500 × 60 × 180 double]
y: [1 × 180 double]
```

The MI classification EEG images extracted with QR-EFA were classified for C3D3a_4C. The data augmentation using Gaussian distribution was used for sufficient training of the ML model and prevention of overfitting. From 10,800 augmented training images, which come from the data augmentation of the original QR neuro images, in brightness and without data augmentation, a total of 2592 non-augmented test images were used. As a result of the final classification experiment, the accuracy was obtained for the test dataset of 2592 sheets, as shown in Table 2.

Table 2. EEG analysis accuracy for C3D3a_4C using QR-EFA method.

		Subjects			
		A1	**A2**	**A3**	**Average**
Accuracy (%)	EFA_LDA, two classes	52.22	46.67	63.33	54.07
	QR-EFA	90.00	93.33	90.83	91.11

Figure 7 shows the examples of eigenfaces, formulated based on the QR code. Although there are lots of QR-eigenfaces are available, we will confine only a small number of eigenfaces, considering the number of classes and trials among the subjects. We chose the eigefacecs reflecting the number of classes. We designed the first to left, second, right, the third to feet, and the last to tongue classes. However, the order is subject to change as the supervisory learning constrictions. This type of decision is related to the choice regarding the approach between the supervised and unsupervised learning.

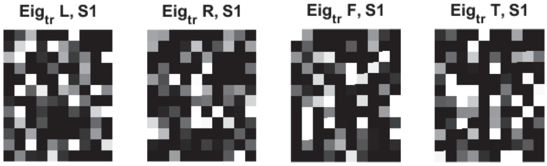
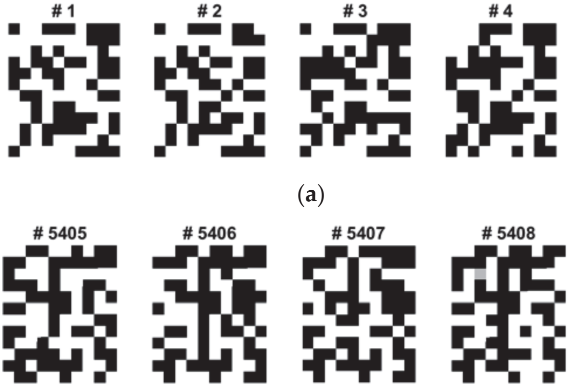


Figure 7. An original eigenface images of subjects for training data for 4 classes: left, right, feet, and tongue after whitening (12 × 10: = 30 × 4) in C3D3a_4C.

Thus far, we obtained the training coefficients by projecting the training images to the QR-eigenfaces and testing the coefficients by projecting the testing images to the QR-eigenfaces. The next step is to conceive of new QR images for training and testing the dataset. Subsequently, linear combinations of the QR eigenfaces, weighted by the relevant coefficients, were used as QR images for training and testing data.

During the data augmentation process for overcoming the limited dataset problem in the BCI competition, we modified the brightness of the recovered images based on the Gaussian noise contamination equation. Although the grayscale and quantized images from the first to last appear similar, the real binary values of the images are different and unique, and they can be used for classification. Because of ML or AI paradigm considerations, a sufficiently large number of training images are needed to train CNNs and fine-tune the logic. However, as the BCI training dataset and its QR images are finite and limited, we need to amplify or diversify them by utilizing data augmentation in the brightness direction. Figure 8 shows one of the EFA QR code implementation results after the brightness data augmentation. Among four classes, considering its dominance and the activation region of the brain, we only selected and data augmented four QR training images for the left and tongue classes, as shown.



(**a**)

(**b**)

Figure 8. Some selected sample QR training images of data augmentation for the class of (**a**) left and (**b**) tongue (12 × 10) in C3D3a_4C.

Because the BCI data are random signals, they must be treated in statistical signal-processing domains. Figure 9 shows the typical neuro raw images considered for the QR-EFA. The left image is for the first trial, and the right image is for the last trial.

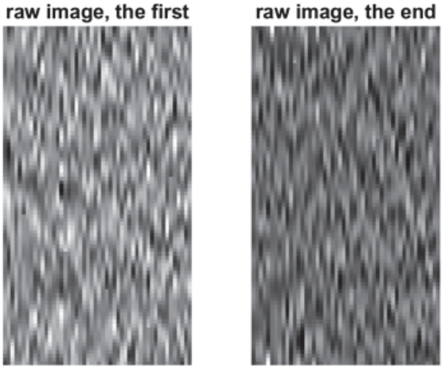


Figure 9. Primitive raw images for EEG data signal for the selected first and last trials (80 × 60) in C3D3a_4C.

Based on the QR-eigenfaces for left, right, feet, and tongue, we proposed considering domain-, trial-, and subject-wise similarities. The detailed considerations are as follows. First, subject-wide similarities among the subjects were checked. Second, domain-wise similarity between the training and testing domains was considered. Finally, a trial-wise similarity was investigated among the trials.

Figure 10 shows the domain-wise similarity between the training and testing domains for subject 1. Considering their importance and the degree of information, only the QR images for subject 1 and trial 1 are shown among training or testing data and several trials. There are many similarities between training and testing QR images. We observed the trial-wise similarity of QR images between the first and last trials for subject 1. We also examined subject-wise similarity of QR images between subjects in the same first trial. It was confirmed that QR-EFA maximizes the three proposed similarities in domain, trials, and subjects.

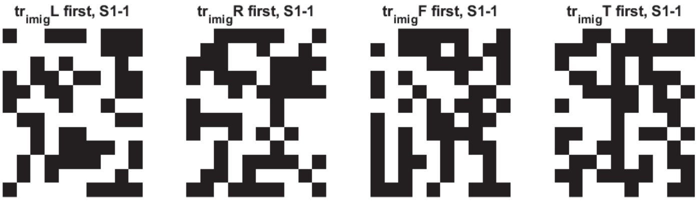


Figure 10. Training QR code realization of subject 1 at trial 1 for 4 classes: left, right, feet, and tongue (12 × 10) in C3D3a_4C.

Using QR-EFA, a sample output of the CNN testing results for BCI competition III dataset 3a (C3D3a_4C) is shown in Figure 11. As the number of epochs increases during simulations, the cost or loss function decreases and reaches a limit.

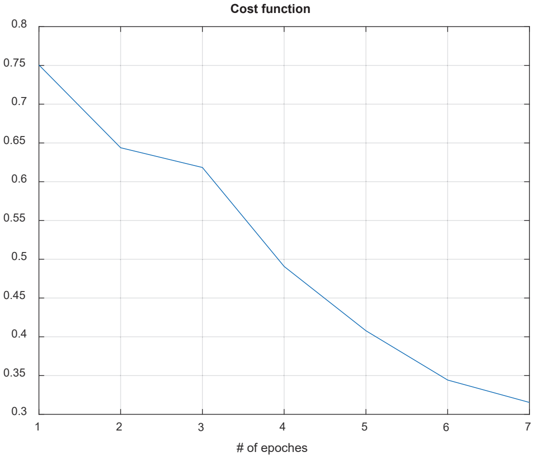


Figure 11. Data graph of the CNN cost function evolution vs. number of epochs in C3D3a_4C.

The MATLAB code for a sample mean of C3D3a_4C data and its confidence interval, when the sample size n = 10 is, as follows (Box 3).

The computation, simulation, verification, and validation on the proposed scheme requires considerations of the confidence interval for the given number of trials (n) and variance or standard deviations. Based on the nature of random seeds in statistical signal processing, we consider a 95% confidence interval, due its variance and spread for data augmentation, CNN layer initiations for kernel filters, and QR-eigenface selections for training images.

Box 3. The MATLAB code for a sample mean of C3D3a_4C.

```
% result set #1
x1(1) = 0.907143; x1(2) = 0.795238; x1(3) = 0.802381; x1(4) = 0.980952;
x1(5) = 0.919048; x1(6) = 0.859524; x1(7) = 0.921429; x1(8) = 0.904762;
x1(9) = 0.909524; x1(10) = 0.821429;
% result set #2
x1(1) = 0.857143; x1(2) = 0.930952; x1(3) = 0.928571; x1(4) = 0.926190;
x1(5) = 0.945238; x1(6) = 0.864286; x1(7) = 0.871429; x1(8) = 0.783333;
x1(9) = 0.928571; x1(10) = 0.952381;
SD1 = std(x1); % Standard deviation (SD)
SE1 = SD1/sqrt(length(x1)); % Standard error(SE)
ts1 = tinv([0.025 0.975],length(x1)−1);
% T-value/score for 95% CI (2.26)
CI1 = mean(x1) + ts1 * SE1;
% Confidence Intervals
>>mean(x1)
Ans = 0.8821
>>std(x1)
ans = 0.0602
>>CI1 % Confidence_Interval
CI1 = 0.8390 0.9252
>>CI1(2)-mean(x1) %Confidence Interval
Ans = 0.0431
```

From the above results, with a 95% confidence interval, the sample mean was estimated to be 0.88, with a confidence interval of 0.0431. Note that, in this case, the Student's *t*-distribution is applicable for calculating the confidence interval and sample variance (s^2), because we do not have information regarding the variance or standard deviation (SD) of the population distribution in BCI datasets (C3D3a_4C or C4D2a_4C both). Table 3 shows the 10 simulation results, as the random seeds changes uniformly for statistically justification for confidence interval.

Table 3. Accuracy results of classification for 10 simulations, with changing random seeds uniformly in C3D3a_2C.

# of Simulations	**Subject (Failed/Trials)**			**Success Rate**
	1	**2**	**3**	
1	26/180	15/120	19/120	0.857143
2	16/180	10/120	3/120	0.930952
3	11/180	13/120	6/120	0.928571
4	9/180	17/120	5/120	0.926190
5	5/180	12/120	6/120	0.945238
6	30/180	14/120	13/120	0.864286
7	27/180	9/120	18/120	0.871429
8	42/180	32/120	17/120	0.783333
9	10/180	15/120	5/120	0.928571
10	18/180	0/120	2/120	0.952381

To calculate a 95% confidence interval for the sample mean μ, an estimated coefficient value 2.26 was used [35]. Thus, the mean and confidence interval of accuracy were 88.21% ± 6.02%, with the confidence interval of 4.31, i.e., (83.90, 92.52). Consequently, the probability for the same mean under the 95% confidence interval is given by Equation (8). For a comparison, the latest and best classification accuracy reported thus far for EEG four classes on BCI competition IV 2a was 85.4 [31].

$$P\left(\left|\overline{X}-\mu\right| < 2.26\sigma/\sqrt{n}\right) = P\left(\left|\frac{\overline{X}-\mu}{s/\sqrt{n}}\right| < 2.26\right) = 0.95 \tag{8}$$

Then, we compute the confidence interval using Equation (9).

$$\left[\overline{X} - 2.26\frac{s}{\sqrt{n}}, \overline{X} + 2.26\frac{s}{\sqrt{n}}\right] = [83.90, 92.52] \quad (9)$$

To validate and verify QR-EFA in a real dataset, with a comparison to C3D3a_4C, the next section is for the result of C4D2a_4C. Table 3 shows the number of trials per subjects for C4D2a_4C. The C4D2a_4C dataset is composed of nine subjects and the predefined number of experimental trials. The number of trials for left, right, foot, and tongue classes in the C4D2a_4C dataset, composed of nine subjects and the predefined number of experimental trials, was 72.

The detailed information of the property for C4D2a_4C is given as follows (Box 4).

Box 4. The detailed information of the property for C3D2a_4C on MATLAB.

```
comment1: 'dataset: C4D2a_4C'
date: '2021.02.12'
madeby: '4C'
affiliation: 'KNIT'
window: 'offset: 3.5, length: 2'
subject: 'subject #: 1'
prefiltering: 'off'
s: 250 (sample/sec)
y: [1 × 288 double]
x: [500 × 22 × 288 double]
c: [22 × 1 cell]
```

Using the designated CNN model, which is depicted in Figure 7, MI classification EEG images extracted with QR-EFA were classified for C4D2a_4C. The accuracy was obtained for the test dataset of 2592 sheets, as shown in Table 4.

Table 4. EEG analysis accuracy for C4D2a_4C, using QR-EFA method.

		Subjects									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Average**
Accuracy (%)	EFA_LDA, two classes	53.47	52.08	55.55	55.55	54.16	45.13	58.33	47.72	51.38	52.55
	QR-EFA	97.15	97.99	97.05	98.33	98.47	97.85	98.78	97.36	97.81	97.87

In Figure 12, the eigenfaces has been formulated. The eigenfaces reflecting the number of classes were selected. Considering a supervisory learning limitation, we designed the first to left, second, right, third to feet, and last to tongue classes.

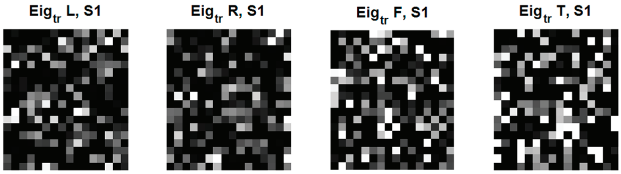


Figure 12. An original eigenface images of subjects for training data for 4 classes: left, right, feet, and tongue after whitening (18 × 16: = 72 × 4) in C4D2a_4C.

The training coefficients by projecting the training images to the QR-eigenfaces and testing coefficients by projecting the testing images to the QR-eigenfaces were obtained. The next step is to conceive of new QR images for the training and testing dataset. Afterwards, we will perform a liner combination of the QR-eigenface, depending on the relevant coefficients. The results of the linear combination were the QR images for the training and testing data. Figure 13 shows the results after brightness data augmentation in the brightness direction. Among four classes, the selected and data augmented images for left and tongue classes are shown.

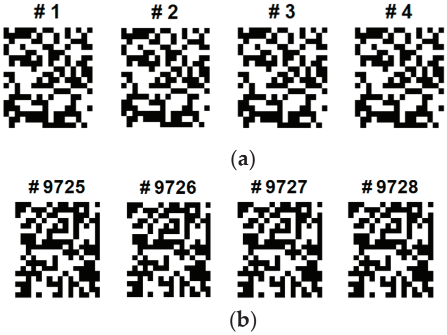


Figure 13. Some selected sample QR training images of data augmentation for the classes of (**a**) left and (**b**) tongue (18 × 16 in C4D2a_4C).

Figure 14 shows the typical neuro raw images. The left image was of the first trial and the right was of the last trial for subject. First, we need to check the subject-wide similarity among subjects. Secondly, we check the domain-wise similarity between training and testing domain. Finally, we check the trial-wise similarity among trials.

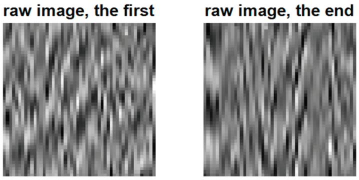


Figure 14. Primitive raw images for EEG data signal for the selected first and last trials (50 × 50) in C4D2a_4C.

Figure 15 shows the domain-wise similarity between training and testing domain for subject 1 and trial 1. There were also similarities between the training and testing images. Hence, we also maximized the similarities in the directions of domain, trials, and subjects.

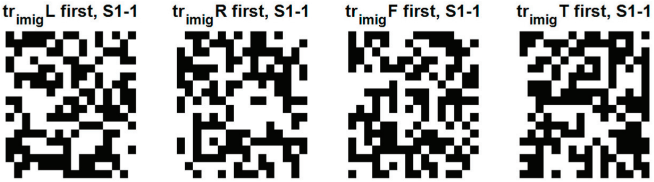


Figure 15. Training QR code realization of subject 1 at trial 1 for 4 classes: left, right, feet, and tongue (18 × 16) in C4D2a_4C.

In summary, it is clear that QR-EFA guarantees a distinguishable and discriminative feature per class, as well as a higher similarity, with respect to domain-, trial-, and subject-wise similarities. This unique novel scheme accelerates the improved accuracy in CNNs. With QR-EFA, a sample output of CNN testing results for in C4D2a_4C is provided in Figure 16. It also shows that the cost function reached the limit.

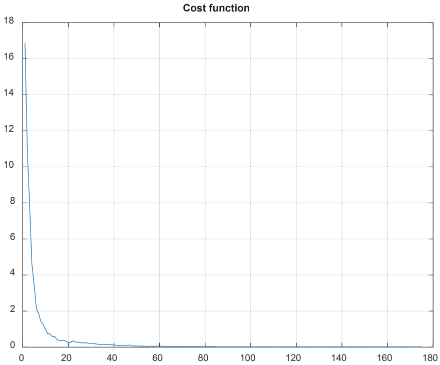


Figure 16. Data graph of the CNN cost function evolution vs. number of epochs in C4D2a_4C.

The MATLAB code for a sample mean of in C4D2a_4C data and its confidence interval, when the sample size n = 10 is, as follows (Box 5).

Box 5. The MATLAB code for a sample mean of in C4D2a_4C data.

```
% result set #1
x1(1) = 0.966435; x1(2) = 0.976852; x1(3) = 0.984954; x1(4) = 0.971065;
x1(5) = 0.969136; x1(6) = 0.983025; x1(7) = 0.986883; x1(8) = 0.985725;
x1(9) = 0.984182; x1(10) = 0.978395;
SD1 = std(x1); % Standard deviation (SD)
SE1 = SD1/sqrt(length(x1)); % Standard error(SE)
ts1 = tinv([0.025 0.975],length(x1)−1); % T-value/score for 95% CI (2.26)
CI1 = mean(x1) + ts1 * SE1; % Confidence Intervals
>>mean(x1)
Ans = 0.9787
>>std(x1)
ans = 0.0075
>>CI1 % Confidence_Interval
CI1 = 0.9733 0.9840
>>CI1(2)-mean(x1) %Confidence Interval
Ans = 0.0054
```

Table 5 shows the 10 simulation results, as the random seeds changes uniformly for statistically justification for confidence interval.

Table 5. Accuracy results of classification for 10 simulations, with changing random seeds uniformly in C4D2a_2C.

# of Simulations	**Subject (Failed/Trials)**									**Success Rate**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
1	18/288	6/288	19/288	9/288	7/288	6/288	2/288	13/288	7/288	0.966435
2	7/288	9/288	3/288	5/288	4/288	8/288	6/288	6/288	12/288	0.976852
3	2/288	2/288	10/288	1/288	6/288	5/288	0/288	4/288	9/288	0.984954
4	19/288	8/288	10/288	7/288	1/288	9/288	7/288	8/288	6/288	0.971065
5	11/288	5/288	10/288	13/288	10/288	12/288	5/288	10/288	4/288	0.969136
6	6/288	4/288	7/288	5/288	3/288	2/288	2/288	8/288	7/288	0.983025
7	3/288	5/288	3/288	1/288	3/288	7/288	4/288	6/288	2/288	0.986883
8	3/288	4/288	5/288	0/288	3/288	4/288	7/288	5/288	6/288	0.985725
9	1/288	8/288	9/288	3/288	3/288	4/288	0/288	9/288	4/288	0.984182
10	12/288	7/288	9/288	4/288	4/288	5/288	2/288	7/288	6/288	0.978395

From the above results, with a 95% confidence interval, we obtained a range of estimates for a sample mean (0.98), and the confidence interval was 0.0054. The mean and confidence interval of the accuracy was 97.87 ± 0.0075%, with a confidence interval of 0.0054, i.e., (97.33, 98.40). The latest and best classification result is shown below. Then, we compute the confidence interval using Equation (10).

$$\left[\overline{X} - 2.26\frac{s}{\sqrt{n}}, \overline{X} + 2.26\frac{s}{\sqrt{n}}\right] = [97.33, 98.40] \tag{10}$$

5. Conclusions

In this study, we proposed the QR-EFA method for efficient motor-imaginary (MI) EEG classification and showed that the EEG signal, when converted into a standardized and sharable QR image, can be classified when using a simple CNN. To obtain a unique feature per class, EEG signal data can be utilized with the concepts of domain-, trial-, and subject- similarities. The EEG data measured from several subjects were used using the BCI competition 3 dataset 3a (C3D3a_4C) and BCI competition 4 dataset 2a (C4D2a_4C). Through QR-EFA method, the EEG signal was converted into EEG QR image, as formed with the EFA method, and a data augmentation technique was applied to solve the limited EEG image problem.

For optimum and best classification performance, QR-EFA maximizes the proposed domain-, trial-, and subject-wise similarities. As far as we are concerned, none of the literature in BCIs has considered the domain-, trial-, and subject-wise similarities so far. Based on this observation, our proposed QR-EFA method was used to maximize the three similarities in the directions of domains, trials, and subjects. Using a simple seven-layer CNN model, data classification results showed an exceptional and remarkable accuracy of 97.87% ± 0.75, with a confidence interval of 0.54, i.e., (97.33, 98.40) for the C4D2a_4C and 88.21% ± 6.02, with a confidence interval of 4.31, i.e., (87.90, 92.52) for the C3D3a_4C. Unlike the CSP method, which is only robust in the original two-class classification, the proposed QR-EFA method is applicable to the classification of two multi-classes of EEG signal; hence, it definitely advantages from the use of the QR-EFA algorithm. Because the QR-EFA method extracts classification features and generates EEG QR images that are well-differentiated between classes, it is suitable as training and testing input images for further ML process and can greatly contribute to classification performance improvement.

Because the ML technique for image classification is becoming stronger with the development of better performance models and hardware performance, QR-EFA's flexible EEG conversion method or frameworks, which should be applied to non-motor imaginary (MI), such as word thinking, emotion detections, arithmetic calculation operations, and multi-class classification, will be developed in the future. The application to a larger number of classes, such as five or six categories, will be applied, with slight extensions in time.

Author Contributions: Conceptualization, H.C., J.P. and Y.-M.Y.; methodology, J.P. and Y.-M.Y.; formal analysis, H.C. and Y.-M.Y.; writing—original draft preparation, H.C. and Y.-M.Y.; supervision, Y.-M.Y. 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 data presented in this study are included within the article.

Acknowledgments: The work reported in this paper was conducted during the sabbatical year of Kumoh National Institute of Technology in 2021. This work was supported by the Gachon University research fund of 2021 (GCU-202110180001).

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

Abbreviations

The following abbreviations are used in this manuscript:

EFA	Eigenface analysis
QR-EFA	Quick eigenface analysis
BCI	Brain computer interface
EEG	Electroencephalogram
CSP	Common spatial pattern
C3D3a_4C	BCI competition III dataset IIIa for four classes
C4D2a_4C	BCI competition IV dataset IIa for four classes
PCA	Principal component analysis
ICA	Independent component analysis

References

1. Classification Ranking for EEG 4 Classes on BCI Competition IV 2a. Available online: https://paperswithcode.com/sota/eeg-4-classes-on-bci-competition-iv-2a (accessed on 22 April 2022).
2. Pfurtscheller, G.; Neuper, C.; Guger, C.; Harkam, W.; Ramoser, H.; Schlogl, A.; Obermaier, B.; Pregenzer, M. Current trends in Graz brain-computer interface (BCI) research. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2000**, *8*, 216–219. [CrossRef] [PubMed]
3. Schalk, G.; McFarland, D.J.; Hinterberger, T.; Birbaumer, N.; Wolpaw, J.R. BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 1034–1043. [CrossRef] [PubMed]
4. Belwafi, K.; Gannouni, S.; Aboalsamh, H. An Effective Zeros-Time Windowing Strategy to Detect Sensorimotor Rhythms Related to Motor Imagery EEG Signals. *IEEE Access* **2020**, *8*, 152669–152679. [CrossRef]
5. Nam, C.S.; Nijholt, A.; Lotte, F. *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*; CRC Press: Boca Raton, FL, USA, 2018.
6. Boi, F.; Moraitis, T.; De Feo, V.; Diotalevi, F.; Bartolozzi, C.; Indiveri, G.; Vato, A. A bidirectional brain-machine interface featuring a neuromorphic hardware decoder. *Front. Neurosci.* **2016**, *10*, 563. [CrossRef]
7. Chin-Teng, L.; Yu-Chieh, C.; Teng-Yi, H.; Tien-Ting, C.; Li-Wei, K.; Sheng-Fu, L.; Hung-Yi, H.; Shang-Hwa, H.; Jeng-Ren, D. Development of Wireless Brain Computer Interface with Embedded Multitask Scheduling and its Application on Real-Time Driver's Drowsiness Detection and Warning. *IEEE Trans. Biomed. Eng.* **2008**, *55*, 1582–1591.
8. Nicolas-Alonso, L.F.; Gomez-Gil, J. Brain Computer Interfaces, a Review. *Sensors* **2012**, *12*, 1211–1279. [CrossRef]
9. Jinyi, L.; Yuanqing, L.; Hongtao, W.; Tianyou, Y.; Jiahui, P.; Feng, L. A Hybrid Brain Computer Interface to Control the Direction and Speed of a Simulated or Real Wheelchair. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2012**, *20*, 720–729.
10. Lemm, S.; Blankertz, B.; Curio, G.; Muller, K. Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Trans. Biomed. Eng.* **2005**, *52*, 1541–1548. [CrossRef]
11. Aznan, N.K.N.; Yeon-Mo, Y. Applying Kalman filter in EEG-Based Brain Computer Interface for Motor Imagery classification. In Proceedings of the 2013 International Conference on ICT Convergence (ICTC), Jeju, Korea, 14–16 October 2013; pp. 688–690.
12. Lotte, F.; Congedo, M.; Lécuyer, A.; Lamarche, F.; Arnaldi, B. A review of classification algorithms for EEG-based brain–computer interfaces. *J. Neural Eng.* **2007**, *4*, R1. [CrossRef]
13. Aznan, N.K.N.; Huh, K.-M.; Yang, Y.-M. EEG-based motor imagery classification in BCI system by using unscented Kalman filter. *Int. J. Inf. Commun. Technol.* **2016**, *9*, 492–508. [CrossRef]
14. Roijendijk, L.; Gielen, S.; Farquhar, J. Classifying Regularized Sensor Covariance Matrices: An Alternative to CSP. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2016**, *24*, 893–900. [CrossRef] [PubMed]
15. Robinson, N.; Vinod, A.P.; Kai Keng, A.; Keng Peng, T.; Guan, C.T. EEG-Based Classification of Fast and Slow Hand Movements Using Wavelet-CSP Algorithm. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 2123–2132. [CrossRef] [PubMed]
16. Husain, A.M.; Sinha, S.R. *Continuous EEG Monitoring: Principles and Practice*; Springer: Berlin/Heidelberg, Germany, 2017.
17. Jin, J.; Xiao, R.; Daly, I.; Miao, Y.; Wang, X.; Cichocki, A. Internal feature selection method of CSP based on L1-norm and Dempster–Shafer theory. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4814–4825. [CrossRef] [PubMed]
18. Blankertz, B.; Muller, K.; Curio, G.; Vaughan, T.M.; Schalk, G.; Wolpaw, J.R.; Schlogl, A.; Neuper, C.; Pfurtscheller, G.; Hinterberger, T.; et al. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 1044–1051. [CrossRef] [PubMed]
19. Vidaurre, C.; Krämer, N.; Blankertz, B.; Schlögl, A. Time Domain Parameters as a feature for EEG-based Brain–Computer Interfaces. *Neural Netw.* **2009**, *22*, 1313–1319. [CrossRef]
20. Lotte, F.; Guan, C. Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms. *IEEE Trans. Biomed. Eng.* **2010**, *58*, 355–362. [CrossRef]
21. Choi, H.; Park, J.; Lim, W.; Yang, Y.-M. Active-beacon-based driver sound separation system for autonomous vehicle applications. *Appl. Acoust.* **2021**, *171*, 107549. [CrossRef]
22. Yu, X.; Chum, P.; Sim, K.-B. Analysis the effect of PCA for feature reduction in non-stationary EEG based motor imagery of BCI system. *Optik* **2014**, *125*, 1498–1502. [CrossRef]

23. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
24. Lu, H.; Eng, H.-L.; Guan, C.; Plataniotis, K.N.; Venetsanopoulos, A.N. Regularized common spatial pattern with aggregation for EEG classification in small-sample setting. *IEEE Trans. Biomed. Eng.* **2010**, *57*, 2936–2946.
25. Bengio, Y.; Goodfellow, I.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017.
26. Wang, Y.; Huang, G.; Song, S.; Pan, X.; Xia, Y.; Wu, C. Regularizing deep networks with semantic data augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3733–3748. [CrossRef] [PubMed]
27. Lee, J.; Won, K.; Kwon, M.; Jun, S.C.; Ahn, M. CNN with large data achieves true zero-training in online P300 brain-computer interface. *IEEE Access* **2020**, *8*, 74385–74400. [CrossRef]
28. The BCI Competition IV Dataset 2a for Four Classes (C4D2a_4C). BCI-Competition III (2008). Available online: https://www.bbci.de/competition/iv (accessed on 25 July 2022).
29. The BCI Competition III Dataset 3a for Four Classes (C3D3a_4C). BCI-Competition III (2005). Available online: https://www.bbci.de/competition/iii (accessed on 25 July 2020).
30. He, L.; Hu, D.; Wan, M.; Wen, Y.; von Deneen, K.M.; Zhou, M. Common Bayesian network for classification of EEG-based multiclass motor imagery BCI. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2016**, *46*, 843–854. [CrossRef]
31. Kim, K.M.; Choe, S.-H.; Ryu, J.-M.; Choi, H. Computation of Analytical Zoom Locus Using Padé Approximation. *Mathematics* **2020**, *8*, 581. [CrossRef]
32. Skansi, S. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2018.
33. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef] [PubMed]
34. Riaz, H.; Park, J.; Choi, H.; Kim, H.; Kim, J. Deep and Densely Connected Networks for Classification of Diabetic Retinopathy. *Diagnostics* **2020**, *10*, 24. [CrossRef]
35. Jung, U.; Choi, H. Active echo signals and image optimization techniques via software filter correction of ultrasound system. *Appl. Acoust.* **2022**, *188*, 108519. [CrossRef]

sensors

Article

Low-Light Image Enhancement Using Hybrid Deep-Learning and Mixed-Norm Loss Functions

JongGeun Oh and Min-Cheol Hong *

School of Electronic Engineering, Soongsil University, Seoul 156-743, Korea
* Correspondence: mhong@ssu.ac.kr; Tel.: +82-2-820-0716

Abstract: This study introduces a low-light image enhancement method using a hybrid deep-learning network and mixed-norm loss functions, in which the network consists of a decomposition-net, illuminance enhance-net, and chroma-net. To consider the correlation between R, G, and B channels, YCbCr channels converted from the RGB channels are used for training and restoration processes. With the luminance, the decomposition-net aims to decouple the reflectance and illuminance and to train the reflectance, leading to a more accurate feature map with noise reduction. The illumination enhance-net connected to the decomposition-net is used to enhance the illumination such that the illuminance is improved with reduced halo artifacts. In addition, the chroma-net is independently used to reduce color distortion. Moreover, a mixed-norm loss function used in the training process of each network is described to increase the stability and remove blurring in the reconstructed image by reflecting the properties of reflectance, illuminance, and chroma. The experimental results demonstrate that the proposed method leads to promising subjective and objective improvements over state-of-the-art deep-learning methods.

Keywords: low-light image enhancement; hybrid deep-learning; mixed-norm; halo artifact; color distortion

Citation: Oh, J.; Hong, M.-C. Low-Light Image Enhancement Using Hybrid Deep-Learning and Mixed-Norm Loss Functions. *Sensors* **2022**, *22*, 6904. https://doi.org/10.3390/s22186904

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 22 August 2022
Accepted: 12 September 2022
Published: 13 September 2022

1. Introduction

The enhancement and miniaturization of image sensors make it possible to easily obtain high-quality images. However, it still remains challenging to overcome external environmental factors, which are the main causes of image degradation and distortion. One such factor, low light, can be a bottleneck in the use of captured images in various applications, such as monitoring, recognition, and autonomous systems [1,2]. A low-light image can be enhanced by adjusting the sensitivity and exposure time of the camera; however, it leads to a blurred image.

Many approaches have been exploited to enhance images in recent decades. Low-light image enhancement can generally be classified into contrast ratio improvement, brightness correction, and cognitive modeling methods. Histogram equalization has been widely used to improve the contrast ratio, and gamma correction has been used to improve image brightness information. However, these methods have limitations in performance improvement because they use arithmetic or statistical methods without considering the illuminance component of an image. Cognitive modeling-based methods correct low illuminance and distorted color signals by dividing the acquired image into illuminance and reflectance components using retinex theory [3]. Single-scale retinex (SSR) [4] and multi-scale retinex (MSR) [5,6] methods have been used to reconstruct low-light images based on retinex theory and random spray [7,8] and illuminance model-based methods [9–12] have been developed as modified versions. Because methods based on the retinex model improve the image by estimating the reflection component, there are problems that cause halo artifacts and color distortion [13]. In addition, variational approaches using optimization techniques have been proposed, but their performance depends on the choice of parameters, and the computational cost is very high [14–16]. Recently, deep-learning-based image processing research has been actively conducted, and various deep-learning methods have been exploited to enhance or reconstruct low-light images [17–22].

Deep-learning-based low-light image restoration methods have advantages and disadvantages depending on their structural characteristics [2]. Most deep-learning methods apply the same architecture to the RGB channels. However, it has been shown that the correlation between the R, G, and B channels is very low; therefore, different architectures suitable to each channel or different color spaces would be more desirable to obtain more satisfactory results. In addition, deep-learning approaches based on the retinex model have been exploited to enhance low-light images. Most of them aim to decouple the reflectance and illuminance components from an input image and enhance only the reflectance [20]. For example, MSR-net using a one-way convolutional neural network (CNN) structure results in color distortion [18]. Retinex-net uses a decomposition neural network (DNN) to decouple the reflectance and illuminance conforming to the retinex-model. However, without considering the different characteristics of the RGB channels, each channel is learned through the same structure, resulting in unstable performance and halo distortion. MBLLEN [21] and KIND [22] attempt to simultaneously control low illuminance and blur distortion using an auto-encoder structure. However, they lead to a loss of detailed image information. Recently, unsupervised learning methods have been reported to solve an over-fitting problem of deep-learning networks on paired images. For example, EnlightGAN uses generator and discriminator models to consider a more realistic environment. Although it leads to promising results, it is tough to train the two models at the same time [23]. In addition, Zero-DCE uses incisive and nonlinear curve mapping [24]. However, unsupervised learning methods have a limitation on performance because a reference image is not used in the loss function.

As described above, deep-learning-based low-light image restoration methods have the problems such as (1) color distortion due to insufficient correlation between color channels and (2) unstable performance and distortion due to the use of the same color channel structure.

To address the above problems, the reflectance and illuminance components are decoupled from the luminance channel of the YCbCr space in this study because the luminance histogram is more similar to the brightness histogram and chrominance channels are less sensitive to additive noise. Based on the converted YCbCr channels, we propose a hybrid structure using decomposition-net, illuminance enhance-net, and chroma-net. The decomposition-net decouples the reflectance and illuminance with the reduction in the additive and shares the weight by extracting the feature map. The illuminance enhance-net connected to the decomposition-net is used to enhance the decoupled illuminance by reducing the halo artifact, which is the main distortion of the retinex-based approaches. In addition, a chroma-net is independently utilized to enhance chroma signals by minimizing color distortion. Moreover, a mixed norm loss function used in each training net is introduced to minimize the instability and degradation of the reconstructed images by reflecting the properties of the reflectance, illuminance, and chroma. The performance of the proposed method is validated using various quantitative evaluators.

The remainder of the paper is organized as follows. Section 2 introduces the proposed deep learning structure and mixed norm-based loss function for low-light image reconstruction. The experimental results and analysis are described in Section 3, and the conclusions are presented in Section 4.

2. Proposed Method

2.1. Hybrid Deep-Learning Structure

The retinex model is the most representative cognitive model for low-light image enhancement, and it can be expressed as follows [3]:

$$S = R \cdot L, \tag{1}$$

where S, R, and L represent the perceptual scene (intensity) of the human eye, reflectance, and illuminance, respectively. Equation (1) is a model that experimentally demonstrates that an object's color varies with ambient illuminance in the human visual system.

This study introduces a hybrid neural network to simultaneously improve illuminance and reflectance components. As mentioned, most deep-learning networks based on the conventional retinex model lead to halo artifacts because they aim to enhance only the reflectance component by decomposing the reflectance and illuminance components from an observed low-light image. Additionally, many deep-learning methods based on the retinex model suffer from color distortion owing to the lack of consideration of the correlation between color channels [25]. To solve these problems, this study adopts a decomposition network that decouples the illuminance and reflectance components and enhances the reflectance in the YCbCr color space. It has been demonstrated that luminance is highly effective in estimating illuminance [26]. Accordingly, illuminance and reflectance are decomposed from the luminance channel, and each channel is used in the training process. In this study, the luminance channel in the YCbCr space is used as an input of the decomposition-net, such that Equation (1) can be rewritten as follows:

$$y = logY = r + l = logR + logL, \tag{2}$$

where Y represents the luminance channel of an observed low-light image, and R and L denote the reflectance and illuminance components of the Y channel, respectively. In addition, the illuminance enhance-net and chroma-net are considered to enhance each component in this study.

The deep-learning structure based on the retinex model should be able to effectively reflect the characteristics of the illumination and reflectance components. In particular, the local homogeneity and spatial smoothness of the illumination component should be effectively decomposed, and the local correlation of the reflectance component should be efficiently extracted [27–29]. Additionally, it is desirable for the network to be capable of removing additive noise.

Figure 1 shows a conceptual diagram of the proposed decomposition network. As shown in the figure, the reflectance and illuminance components decoupled from the luminance channel share weights by extracting feature maps that conform to the model by specifying a loss function for each output. In Figure 1, y_{low}, $\bar{l}_{low}$, and $\bar{r}_{low}$ represent the low-light luminance, trained illuminance, and trained reflectance components, respectively. In contrast, y_{GT}, $\bar{l}_{GT}$, and $\bar{r}_{GT}$ represent the paired ground-truth luminance, trained illuminance, and trained reflectance, respectively.

As shown in Figure 2, the proposed deep-learning network consists of three stages: (1) decomposition-net, (2) illuminance enhance-net, and (3) chroma-net. As previously mentioned, the decomposition-net accurately decouples the reflectance and illuminance components from the luminance channel. In addition, the illuminance enhance-net is used to learn about the illuminance, and chroma-net is added to consider the chroma characteristics.

The proposed decomposition-net considers the characteristics of each component and composites the sub-network structure to facilitate training. Sub-neural networks consist of a forward CNN, an auto encoder-based neural network, and a multi-scale-based neural network using skip-connections. For illuminance, a multi-scale CNN structure including various-sized receptive fields is used to decompose the local homogeneity and spatial smoothness. This structure is capable of obtaining a feature extraction map that is robust to various input images [2]. The reflectance component is easier to preserve and learn detailed information and boundary information of the image than the illumination component. Applying these characteristics, the reflectance component uses a forward small-scale receptive field to facilitate learning the local correlation of the image. The auto-encoder has a structure that combines learning feature maps of different sizes using a skip- connection. It has been shown that this structure is easy to learn through structural analysis of the image and that it is effective in removing additive noise in the image [30].

As described above, to effectively remove the noise present in the low-light image, an auto-encoder structure is used for the reflection component. Figure 3 shows the structural diagrams for the multi-scale CNN (sub-net 1), forward CNN (sub-net 2), and auto-encoder (sub-net 3) used in this study.

The parallel structure described above becomes structurally flexible by learning the decomposition components, thereby shortening the learning time and clarifying the role of each sub-network. In addition, illuminance enhance-net and chroma-net use a forward CNN to enhance each component because illuminance and chroma include less additive noise than reflectance, such that over-blurring can be avoided.

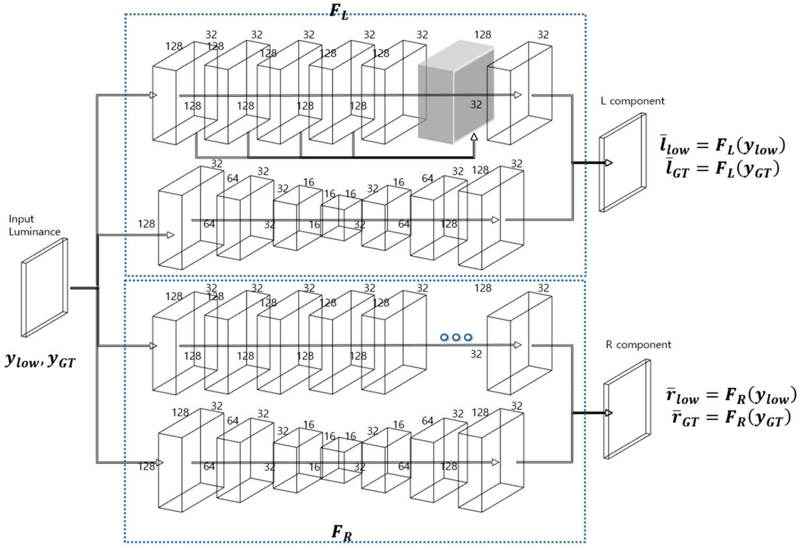

F_L
Input Luminance
y_{low}, y_{GT}
L component
$\bar{l}_{low} = F_L(y_{low})$
$\bar{l}_{GT} = F_L(y_{GT})$
R component
$\bar{r}_{low} = F_R(y_{low})$
$\bar{r}_{GT} = F_R(y_{GT})$
F_R
128
32
64
16

Figure 1. Conceptual diagram of proposed decomposition network.

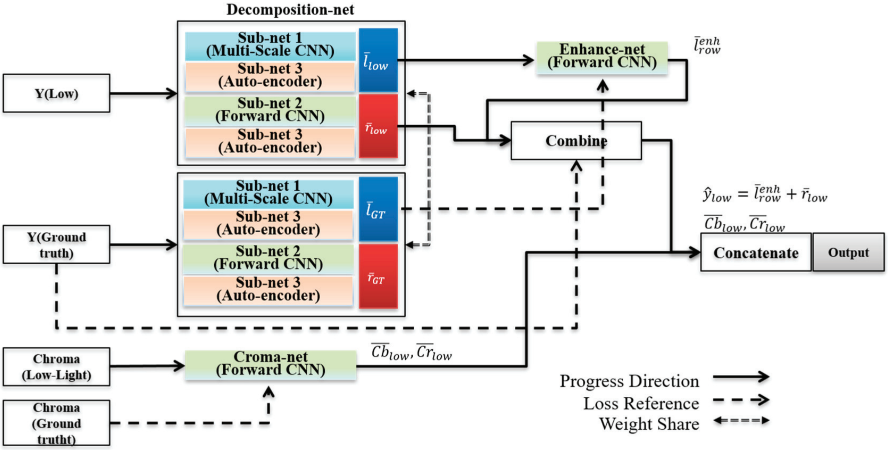


Figure 2. Flowchart of proposed network.

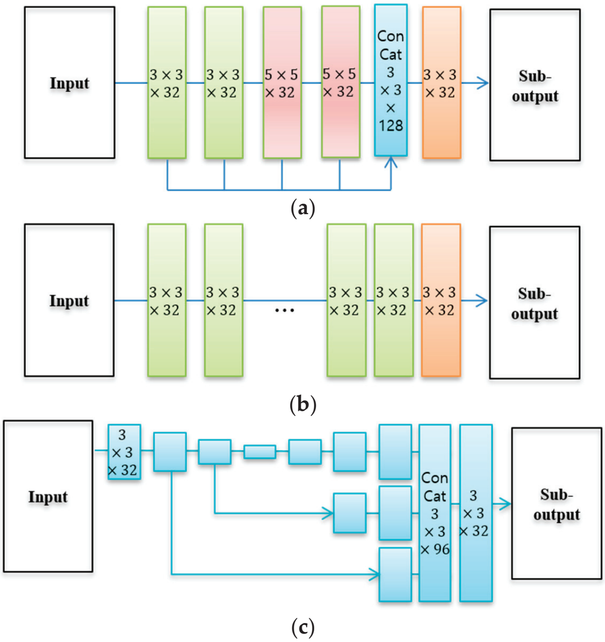


Figure 3. Architectures of sub-network: (**a**) multi-scale CNN (sub-net 1), (**b**) forward CNN (sub-net 2), (**c**) auto-encoder (sub-net 3).

2.2. Mixed Norm-Based Loss Function

A loss function using the hybrid learning architecture is defined to effectively train the input pair by minimizing the error of each learning system, i.e., the decomposition-net, illuminance enhance-net, and chroma-net.

The decomposition-net accurately extracts the reflectance from the luminance, and the loss function is defined as follows:

$$Loss^{D} = L_d + L_r + L_l, \tag{3}$$

where L_d, L_r, and L_l represent the decomposition, reflectance, and illuminance loss functions, respectively.

The decomposition loss function is a basic loss function using the retinex model and can be written as follows:

$$L_d = \|\bar{r}_{low} - \bar{r}_{GT}\|_1 + \alpha_1\|\bar{r}_{low} + \bar{l}_{GT} - \widetilde{y}_{GT}\|_2^2 + \alpha_2\|\bar{r}_{low} + \bar{l}_{low} - \widetilde{y}_{low}\|_2^2, \tag{4}$$

where $\widetilde{y}_{GT}$ and $\widetilde{y}_{low}$ denote the normalized ground truth luminance channel and paired low-light luminance, respectively, in which the elements of $\widetilde{y}_{GT}$ and $\widetilde{y}_{low}$ are scaled to [0, 1]. For an $M \times N$-sized image, each symbol is a lexicographically ordered $MN \times 1$ column vector. In Equation (4), each term represents the model-based loss function in which the first term uses the L1 norm because it includes the details and boundary information of the image.

The reflectance model loss function should contain detailed information regarding the object. Therefore, the minimization terms for the gradient map and the error term for the ground-truth image are included to reflect the property. The loss function for the reflection model is expressed as follows:

$$L_r = \beta_1\|\nabla\bar{r}_{low} - \nabla\widetilde{y}_{GT}\|_1 + \beta_2\|\bar{r}_{low} - \widetilde{y}_{GT}\|_2^2, \tag{5}$$

where ∇ represents the gradient operator, and β_1 and β_2 denote the regularization parameters to control the relative contribution of each term.

In general, illuminance is suitable for representing an object surface as a Lambertian model [14]. Accordingly, the illuminance model loss function can be expressed as follows:

$$L_l = \gamma \left[\frac{\nabla \bar{l}_{low}}{max(\nabla \widetilde{y}_{low}, \varepsilon)} + \frac{\nabla \bar{l}_{GT}}{max(\nabla \widetilde{y}_{GT}, \varepsilon)} \right], \tag{6}$$

where γ is the regularization parameter for the loss function, and ε is a small constant to prevent the denominator from becoming zero.

The loss functions for training the illuminance component and chroma signals are expressed as follows:

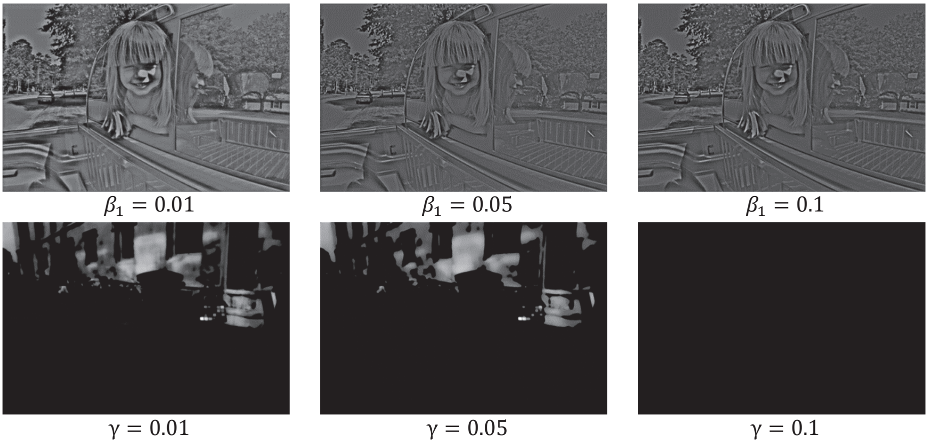
$$Loss^L = \| \bar{l}^{enh}_{row} - \bar{l}_{GT} \|_2^2, \tag{7}$$

where the initial vector is equal to $\bar{l}_{low}$, and

$$Loss^C = \| \overline{C}_{row,i} - \widetilde{C}_{GT,i} \|_2^2 \; i \in \{r, b\}, \tag{8}$$

where i denotes the chrominance channel index and $\widetilde{C}_{GT,i}$ represents the normalized i-th chrominance signal of the ground-truth image. An element of the chrominance takes a value between -128 and 128; thus, it is normalized to [0, 1] for training. The Adam method [31] was used to obtain the optimized solution of the loss functions, and 55 batch sizes were applied in the training process.

As expressed above, the proposed decomposition loss function consists of various functions; therefore, the convergence of the loss function depends on the choice of the parameters. The selection of optimized parameters is beyond the scope of this work. In this study, these parameters were experimentally determined. The regularization parameters (α_1 and α_2) in Equation (4) are due to the retinex theory, and when they have low values, the decomposition-net fails to extract an accurate feature map. It was observed that the decomposition-net satisfactorily converged with $\alpha_1,\ \alpha_2 > 0.1$. In addition, it was confirmed that as β_1 in Equation (5) increases, detailed information, such as boundaries, is well expressed in the feature map of the reflectance component. However, it was experimentally confirmed that the parameter β_2 for preserving the overall structure did not affect the results. Additionally, it was verified that the spatial flatness of the feature map of the illuminance component increased as γ increased. Figure 4 shows an example of the variation in the feature map for various β_1 and γ.



$\beta_1 = 0.01$ $\beta_1 = 0.05$ $\beta_1 = 0.1$

$\gamma = 0.01$ $\gamma = 0.05$ $\gamma = 0.1$

Figure 4. Variation of feature map for various β_1 and γ.

3. Experimental Results

3.1. Experimental Setup

Several experiments were conducted using various low-contrast images. The dataset used for training consisted of a pair of ground-truth and low-light images. Overall, 1300 ground-truth images were selected from the LIVE [32], Google Image-net [33], NASA ImageSet [34], and BSDS500 [35] datasets. The degraded images were generated from the ground-truth images using two random variables. The random variables were as follows:

(1) gamma correction: $\Gamma \in [2.5,\ 3.0]$,

(2) random spray Gaussian noise: random spray ratio (0.01%) and Gaussian std. $\in [35.0,\ 45.0]$.

A total of 6500 degraded images using the variables were randomly generated, and the average spatial resolution of the images was 884 × 654. In this work, we describe the experimental results of 50 real low-light images and distorted images of 50 ground-truth images that were not used for training. In addition, the parameters used in the loss function were set as $\alpha_1 = \alpha_2 = 1.0$, $\beta_1 = \beta_2 = 0.1$, and $\gamma = 0.01$.

The proposed method compares the performance with MSR-net [18], Retienx-net [20], MBLLEN [21], and KinD [22] in terms of various evaluations, such as the peak-to-signal ratio (PSNR), lightness order error (LOE) [36], visual information facility (VIF) [37], perceptual based image quality evaluator (PIQE) [38], structural similarity index measure (SSIM) [39], and contrast per pixel (CPP) [40]. The LOE represents the number of pixels in which the lightness alignment between the reference image and the comparison image within a 50 × 50 window at the reference point deviates. The VIF is an evaluator that determines the degree of improvement or inhibition compared to the reference image using a statistically established index. In addition, PIQE, which does not require a reference image, represents the degree of natural representation of the image, where a smaller value indicates that the image is more natural from a cognitive perspective. On the other hand, CPP represents the amount of change in contrast within a 3 × 3 window, such that there is a limit to the evaluation of image quality improvement. However, it is suitable for evaluating the similarity of the amount of contrast change with the ground-truth image. An Intel E3-1276 3.6 GHz with 32 GB RAM and NVIDIA 1660Ti GPU were used to run the algorithms with the TensorFlow 1.2 library of Python 3.0.

3.2. Analyses of Experimental Results

Table 1 shows the performance comparisons, where ↑ indicates a quantitative improvement as the value increases. The results show that the proposed method outperforms other methods in terms of PSNR, VIF, and PIQE. In particular, the PSNR, SSIM, VIF, and PIQE improved by 1.7~6.2 dB, 0.02~0.13, 0.04~0.2, and 7~10 with respect to the comparative methods, respectively. In contrast, MBLLEN dominates the others with respect to the LOE. Because the LOE evaluates the match between the alignment of the reference image and the corresponding comparative image as on/off, there is a limit to the accuracy of evaluating the performance improvement. In addition, the retinex-net generated halo-artifact, which is one of the main problems of retinex-based methods; further, it was confirmed that this distortion was a factor that increased the CPP value. In addition, it was observed that KinD outperforms the other comparative methods with respect to the PSNR, SSIM, and VIF. However, the LOE is very high due to the halo artifact. Through the results of the quantitative evaluations, it was confirmed that the proposed method reconstructed the image closest to the ground-truth image, and similar results were confirmed for the low-light image without the ground-truth image. In particular, the PIQE comparisons show that the proposed method has the capability to reconstruct more natural images.

Table 1. Performance comparisons (**Blue**: the best, **Red**: the second best).

	Evaluator	Ground Truth	Degraded Image	MSR-Net [18]	Retinex-Net [20]	MBLLEN [21]	KinD [22]	Proposed Method
with reference	PSNR ↑	N/A	8.69	15.88	17.64	19.60	**20.14**	**22.01**
	SSIM ↑	N/A	0.547	0.800	0.766	0.823	**0.873**	**0.897**
	LOE ↓	N/A	282.90	210.94	374.09	**202.57**	327.84	**208.04**
	VIF ↑	N/A	0.366	0.508	0.451	0.556	**0.613**	**0.656**
	PIQE ↓	36.94	39.83	**37.60**	47.47	47.41	51.17	**30.96**
	CPP	35.98	15.07	29.36	**47.50**	25.82	30.03	**31.27**
without reference	PIQE ↓	N/A	33.25	**31.06**	39.25	52.65	46.17	**24.02**
	CPP	N/A	13.93	19.64	**35.66**	14.44	20.01	**20.63**

Visual performance comparisons are shown in Figures 5 and 6. It was observed that MSR-net is not promising with respect to luminance correction and color maintenance because it uses only feedforward training. In addition, the retinex-net has the problem of halo artifacts and color distortion. The halo artifact of the retinex-net is the main factor that increases the LOE and CPP, which agrees with the results shown in Table 1. The results verify that a different structure is applied to each channel due to insufficient correlation between RBG channels. On the other hand, MBLLEN is effective in removing additive noise using the convolutional neural layer based on an auto-encoder structure, but it is confirmed that illuminance improvement is insufficient and over-blurring is caused by over-denoising. KinD resulted in satisfactory illuminance improvement but led to color distortion and halo artifacts in the reconstructed images. However, the proposed method resulted in promising improvements. In particular, the experimental results show that the proposed method more naturally reconstructs the image compared with the other methods through illuminance improvement and color correction. As shown in Figure 6, similar results were obtained with real low-light images having uneven brightness and multi-light sources. Through the experimental results, it is confirmed that brightness correction, color maintenance, noise suppression, and halo artifact reduction should be simultaneously considered in low-light image enhancement. The experiments proved that the hybrid deep-learning structure and mixed-norm loss functions yield subjectively and objectively promising results.

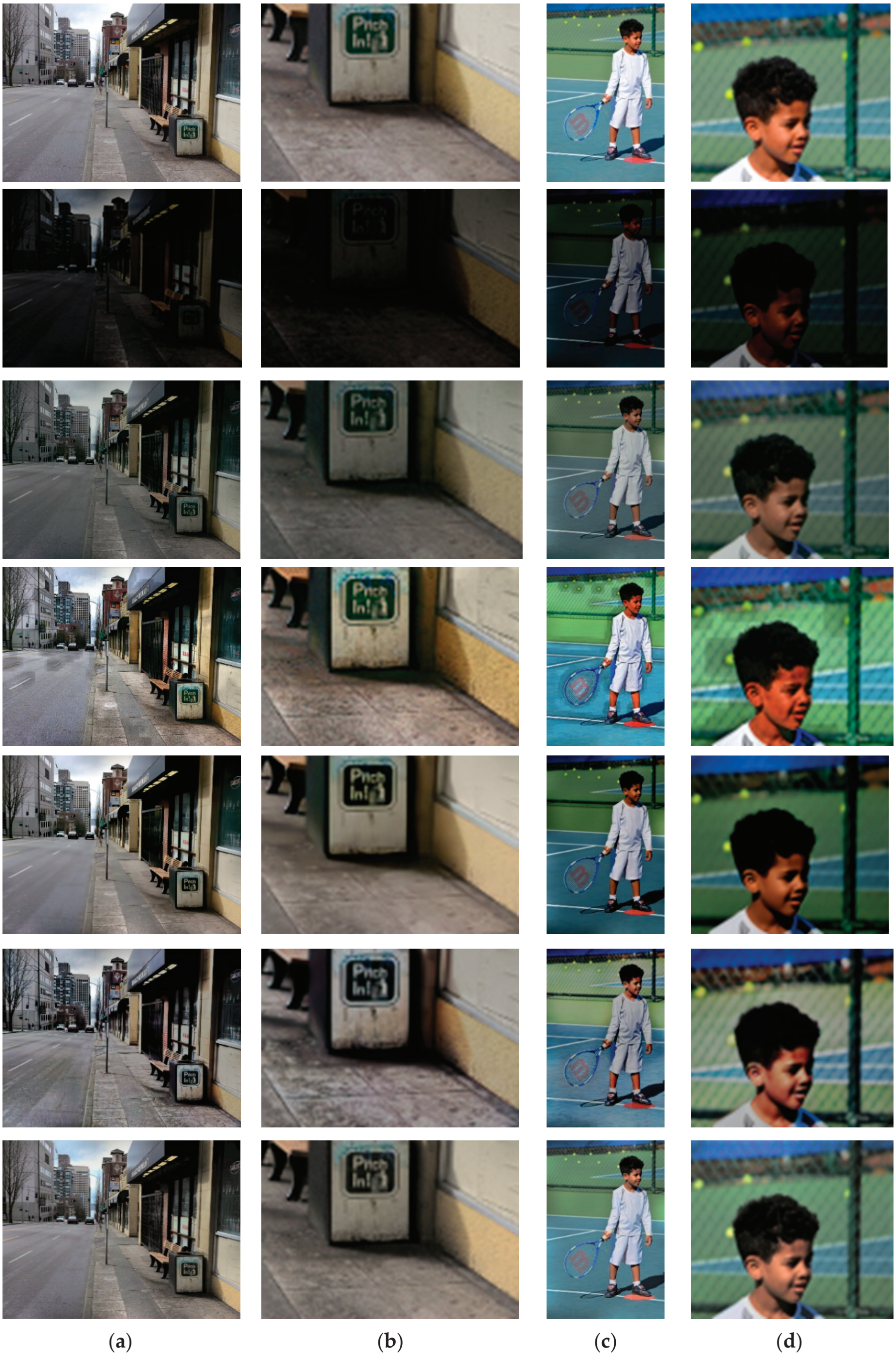


Figure 5. Visual comparisons of with-ground-truth images: (from top to bottom) ground-truth image, degraded image, MSR-net, retinex-net, MBLLEN, KinD, and proposed method. (**a**) test image1, (**b**) partially zoomed-in view of (**a**), (**c**) test image2, and (**d**) partially zoomed-in view of (**c**).

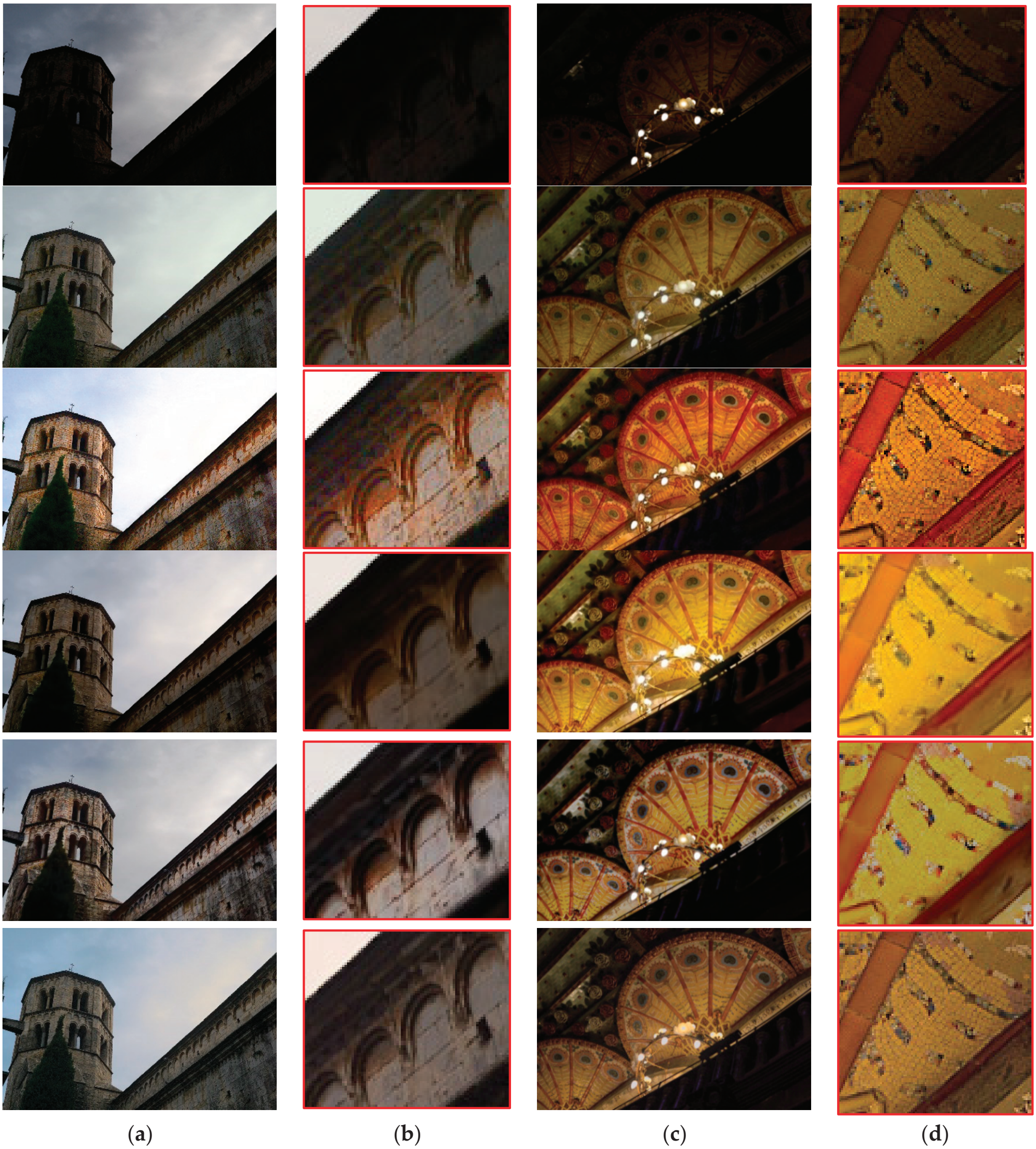


Figure 6. Visual comparisons of without-ground-truth images: (from top to bottom) real low-light image, MSR-net, retinex-net, MBLLEN, KinD, and proposed method. (**a**) test image3, (**b**) partially zoomed-in view of (**a**), (**c**) test image4, and (**d**) partially zoomed-in view of (**c**).

4. Conclusions

This study introduces a hybrid deep-learning network and mixed norm loss functions, in which the hybrid net consists of a decomposition-net, illuminance enhance-net, and chroma-net, each of which is defined to reflect the properties. To improve brightness and reduce halo artifacts and color distortion, the YCbCr channels are used as inputs for the hybrid network. Then, the illuminance and reflectance are decoupled from the luminance channel, and the reflectance is trained by decomposition-net, such that the reflectance is enhanced, and the additive noise is efficiently removed. In addition, an enhance-net connected to the decomposition-net is introduced, resulting in illuminance improvement and reduction in halo artifacts. Moreover, the chroma-net is separately included in the hybrid-net because the properties of chroma channels are different from those of luminance, leading to a reduction in color distortion. In addition, a mixed norm loss function is introduced to minimize the instability and degradation of the reconstructed images by reflecting the properties of the reflectance, illuminance, and chroma.

The experiments confirmed that the proposed method showed satisfactory performance in various quantitative evaluations compared with other competitive deep-learning methods. In particular, it was verified that the proposed method could effectively enhance brightness and reduce additive noise, color distortion, and halo artifacts. It is expected that the proposed method can be applied to various intelligent imaging systems to obtain a high-quality image. Currently, deep-learning methods for low-light videos are under development. The newest methods are expected to reduce flickering artifacts between frames and to achieve even better performance.

Author Contributions: J.O. and M.-C.H. conceived and designed the experiments; J.O. performed the experiments; J.O. and M.-C.H. analyzed the data; M.-C.H. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grant funded by Korean government (MIST) (2020R1A2C1003897).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Anyone can use or modify the source-code for only academic purposes. The source-code will be accessed on 15 October 2022 at https://drive.google.com/drive/folders/153qbJeMO96qSLS6qVr513v7_O8aIuKHI.

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

References

1. Chien, J.-C.; Chen, Y.-S.; Lee, J.-D. Improving night time driving safety using vision-based classification techniques. *Sensors* **2017**, *17*, 10. [CrossRef] [PubMed]
2. Wang, W.; Wu, X.; Yuan, X.; Gao, Z. An experimental-based review of low-light image enhancement methods. *IEEE Access* **2020**, *8*, 87884–87917. [CrossRef]
3. Land, E.; McCann, J. Lightness and retinex theory. *J. Opt. Soc. Am.* **1971**, *61*, 1–11. [CrossRef] [PubMed]
4. Jobson, D.; Woodell, G. Properties and performance of a center/surround retinex. *IEEE Trans. Image Process.* **1997**, *6*, 451–462. [CrossRef]
5. Rahman, Z.; Jobson, D.; Woodell, G. Multi-scale retinex for color image enhancement. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 16–19 September 1996; pp. 1003–1006.
6. Jobson, D.; Rahman, Z.; Woodell, G. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Trans. Image Process.* **1997**, *6*, 965–976. [CrossRef]
7. Provenzi, E.; Fierro, M.; Rizzi, A.; Carli, L.D.; Gadia, D.; Marini, D. Random spray retinex: A new retinex implementation to investigate the local properties of the model. *IEEE Trans. Image Process.* **2007**, *16*, 162–171. [CrossRef]
8. Banic, N.; Loncaric, S. Light random spray retinex: Exploiting the noisy illumination estimation. *IEEE Signal Process. Lett.* **2013**, *20*, 1240–1243. [CrossRef]
9. Celik, T. Spatial Entropy-Based Global and Local Image Contrast Enhancement. *IEEE Trans. Image Process.* **2014**, *23*, 5209–5308. [CrossRef]
10. Shin, Y.; Jeong, S.; Lee, S. Efficient naturalness restoration for non-uniform illuminance images. *IET Image Process.* **2015**, *9*, 662–671. [CrossRef]
11. Lecca, M.; Rizzi, A.; Serapioni, R.P. GRASS: A gradient-based random sampling scheme for Milano retinex. *IEEE Trans. Image Process.* **2017**, *26*, 2767–2780. [CrossRef]
12. Simone, G.; Audino, G.; Farup, I.; Albregtsen, F.; Rizzi, A. Termite retinex: A new implementation based on a colony of intelligent agents. *J. Electron. Imaging* **2014**, *23*, 013006. [CrossRef]
13. Dou, Z.; Gao, K.; Zhang, B.; Yu, X.; Han, L.; Zhu, Z. Realistic image rendition using a variable exponent functional model for retinex. *Sensors* **2017**, *16*, 832. [CrossRef]
14. Kimmel, R.; Elad, M.; Sobel, I. A variational framework for retinex. *Int. J. Comput. Vis.* **2003**, *52*, 7–23. [CrossRef]
15. Zosso, D.; Tran, G.; Osher, S.J. Non-local retinex-A unifying framework and beyond. *SIAM J. Imaging Sci.* **2015**, *8*, 787–826. [CrossRef]
16. Park, S.; Yu, S.; Moon, B.; Ko, S.; Paik, J. Low-light image enhancement using variational optimization-based retinex model. *IEEE Trans. Consum. Electron.* **2017**, *63*, 178–184. [CrossRef]

17. Lore, K.G.; Akintayo, A.; Sarkar, S. LLNet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognit.* **2017**, *61*, 650–662. [CrossRef]
18. Shen, L.; Yue, Z.; Feng, F.; Chen, Q.; Liu, S.; Ma, J. MSR-net: Low-light image enhancement using deep convolutional network. *arXiv* **2017**, arXiv:171102488.
19. Guo, C.; Li, Y.; Ling, H. Lime: Low-light image enhancement via illuminance map estimation. *IEEE Trans. Image Process.* **2017**, *26*, 982–993. [CrossRef]
20. Wei, C.; Wang, W.; Yang, W.; Liu, J. Deep retinex decomposition for low-light enhancement. *arXiv* **2018**, arXiv:180804560.
21. Lv, F.; Lu, F.; Wu, J.; Lim, C. MBLLEN: Low-light image/video enhancement using CNNs. In Proceedings of the British Machine Vision Conference (BMVC), Newcastle, UK, 3–6 September 2018; pp. 1–13.
22. Zhang, Y.; Zhang, J.; Guo, X. Kindling the darkness: A practical low-light image enhancer. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 15 October 2019; pp. 1632–1640.
23. Jiang, Y.; Gong, X.; Liu, D.; Cheng, Y.; Fang, C.; Shen, X.; Yang, J.; Zhou, P.; Wang, Z. EnlightenGAN: Deep light enhancement without paired supervision. *IEEE Trans. Image Process.* **2021**, *30*, 2340–2349. [CrossRef]
24. Guo, C.; Li, C.; Guo, J.; Loy, C.C.; Hou, J.; Kwong, S.; Cong, R. Zero-reference deep curve estimation for low-light image enhancement. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1780–1789.
25. Kim, B.; Lee, S.; Kim, N.; Jang, D.; Kim, D.-S. Learning color representation for low-light image enhancement. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 1455–1463.
26. Oh, J.-G.; Hong, M.-C. Adaptive image rendering using a nonlinear mapping-function-based retinex model. *Sensors* **2019**, *19*, 969. [CrossRef]
27. Kinoshita, Y.; Kiya, H. Convolutional neural networks considering local and global features for image enhancement. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019. [CrossRef]
28. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Shahbaz, F.; Yang, M.-H.; Shao, L. Learning enriched features for real image restoration and enhancement. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 492–511.
29. Anwar, S.; Barnes, N.; Petersson, L. Attention-based real image restoration. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [CrossRef]
30. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **2016**, *26*, 3142–3155. [CrossRef]
31. Kingman, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2017**, arXiv:14126980v9.
32. Sheikh, H.R.; Wang, Z.; Cormack, L.; Bovik, A.C. Live Image Quality Assessment Database Release 2. The Univ. of Texas at Austin. 2005. Available online: https://live.ece.utexas.edu/research/Quality/subjective.htm (accessed on 23 March 2022).
33. Stanford Vision Lab. ImageNet. 2016. Available online: http://image-net.org (accessed on 18 May 2022).
34. NASA Langley Research Center. Available online: https://dragon.larc.nasa.gov (accessed on 17 November 2021).
35. Arbelaez, P.; Fowlkes, C.; Martin, D. The Berkeley Segmentation Dataset and Benchmark. 2007. Available online: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (accessed on 7 February 2022).
36. Wang, S.; Zheng, J.; Hu, H.; Li, B. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE Trans. Image Process.* **2013**, *22*, 3538–3548. [CrossRef]
37. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444. [CrossRef]
38. Venkatanath, N.; Praneeth, D.; Chandrasekhar, B.H.; Channappayya, S.S.; Medasani, S.S. Blind image quality evaluation using perception based features. In Proceedings of the 21st National Conference on Communications (NCC), Mumbai, India, 27 February–1 March 2015. [CrossRef]
39. 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]
40. Peli, E. Contrast in complex images. *J. Opt. Soc. Am. A* **1990**, *7*, 2032–2040. [CrossRef]

sensors

Article

Instance-Level Contrastive Learning for Weakly Supervised Object Detection

Ming Zhang *and Bing Zeng

School of Information and Communication Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu 611731, China

* Correspondence: zm_zhangming@std.uestc.edu.cn

Abstract: Weakly supervised object detection (WSOD) has received increasing attention in object detection field, because it only requires image-level annotations to indicate the presence or absence of target objects, which greatly reduces the labeling costs. Existing methods usually focus on the current individual image to learn object instance representations, while ignoring instance correlations between different images. To address this problem, we propose an instance-level contrastive learning (ICL) framework to mine reliable instance representations from all learned images, and use the contrastive loss to guide instance representation learning for the current image. Due to the diversity of instances, with different appearances, sizes or shapes, we propose an instance-diverse memory updating (IMU) algorithm to mine different instance representations and store them in a memory bank with multiple representation vectors per class, which also considers background information to enhance foreground representations. With the help of memory bank, we further propose a memory-aware instance mining (MIM) algorithm that combines proposal confidence and instance similarity across images to mine more reliable object instances. In addition, we also propose a memory-aware proposal sampling (MPS) algorithm to sample more positive proposals and remove some negative proposals to balance the learning of positive-negative samples. We conduct extensive experiments on the PASCAL VOC2007 and VOC2012 datasets, which are widely used in WSOD, to demonstrate the effectiveness of our method. Compared to our baseline, our method brings 14.2% mAP and 13.4% CorLoc gains on PASCAL VOC2007 dataset, and 12.2% mAP and 8.3% CorLoc gains on PASCAL VOC2012 dataset.

Keywords: weakly supervised object detection; instance-level contrastive learning; memory-aware instance mining; memory-aware proposal sampling

Citation: Ming, Z.; Bing, Z. Instance-Level Contrastive Learning for Weakly Supervised Object Detection. *Sensors* **2022**, *22*, 7525. https://doi.org/10.3390/s22197525

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 2 September 2022
Accepted: 3 October 2022
Published: 4 October 2022

1. Introduction

Object detection is a fundamental task in computer vision, which requires to identify object categories and use bounding boxes to locate their complete region positions. With the development of convolutional neural network (CNN) [1–3], some object detection methods [4–13], such as Fast R-CNN [4], Faster R-CNN [5], SSD [6] and YOLO [7], have made significant progress. However, these methods require fully supervised information, i.e., instance-level annotations, which are time-consuming and labor-intensive to label. To reduce the burden of annotations, weakly supervised object detection (WSOD) removes bounding boxes and only requires image-level annotations, i.e., image tags, to indicate whether object categories are present in an image.

Due to lack of object bounding box position supervision, most current WSOD methods [14–27] use multiple instance learning (MIL) [28] to mine object instances from pre-generated proposals, and treat them as pseudo instance-level annotations to train weakly supervised detectors. However, these methods only focus on a single image to learn object representations without considering the internal relevance of various object instances across images. When there are object appearance variations in the complex diverse image scenes, it is easy to cause false detection. For example, when a horse is occluded in Figure 1, it focuses more on local feature representation, which is not enough to represent the whole object, resulting in the learned object instance only covering the head of the horse.

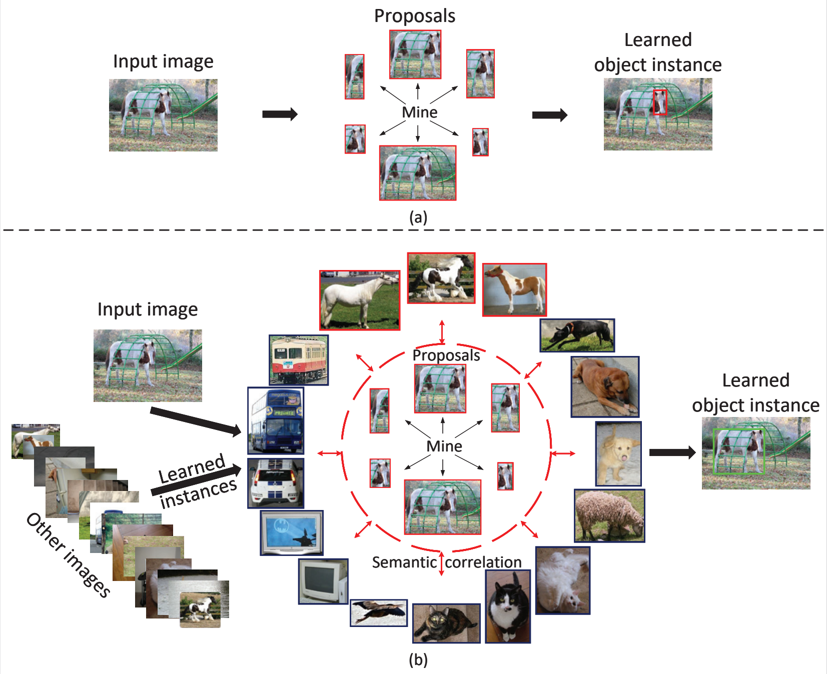


Figure 1. Illustration of our motivation through data flow. (**a**) Most WSOD methods usually learn object instances from region proposals of current input image. (**b**) Our method establishes instance correlations with other images to guide instance representation learning for the current input image.

To deal with the problem, we propose an instance-level contrastive learning (ICL) framework to store reliable instance representations from all learned images, and utilize contrastive learning [29] mechanism to explicitly establish semantic correlations with other image instances. It attempts to enhances the discriminative and robustness of object instance representations in the current input image, pulling it close to the instance representations of same class from all training images and pushing it away instance representations of different classes. As shown in Figure 1b, owning to instance correlation with other images, our method can effectively learn the instance representation for the whole horse.

To sufficiently represent diversity instances in all training data, we next propose an instance-diverse memory updating (IMU) algorithm. It mines reliable instance representations from proposal features and builds a memory bank with multiple representation vectors for each class to store them based on similarity, where background information is also consider to enhance foreground representations. Based on the memory bank, we further propose a memory-aware instance mining (MIM) algorithm. Unlike most methods [14,15,20–22,26,27] that mine object instances only based on proposal confidence, we also compute the similarity with stored diverse instances to evaluate the completeness of proposals to mine more reliable object instances. Instead of selecting the top-scoring proposal as an instance, we also consider the multi-instance case to mine more instances in an image. During the training process of weakly supervised object detectors, we propose a memory-aware proposal sampling (MPS) algorithm to alleviate the imbalance problem between positive and negative samples. According to the similarity with instance representations, we select more positive proposals to increase the number of positive samples. Based on the similarity with the background information, we remove some negative proposals with low similarity to reduce the number of negative samples.

To verify the effectiveness of our method, we conduct extensive experiments on the PASCAL VOC2007 and VOC2012 datasets, which are widely used for weakly supervised object detection. In this paper, we adopt typical WSOD method OICR [15] as our baseline, which can be easily embedded into our ICL framework to further improve the performance. On PASCAL VOC2007 dataset, our method improves detection performance and localization accuracy by 14.2% and 13.4% in terms of mAP and CorLoc, respectively. On PASCAL VOC2012 dataset, our method improves performance by 12.2% mAP and 8.3% CorLoc.

The contributions of this paper are summarized as follows:

- We propose an instance-level contrastive learning (ICL) framework to guide the weakly supervised detector to learn instance representations. To the best of our knowledge, we are the first to explore contrastive learning in weakly supervised object detection.
- We propose an instance-diverse memory update (IMU) algorithm to store reliable instance representations into a memory bank, where multiple representation vectors are used in each class to maintain the diversity of instance representations.
- With the help of memory, we further propose a memory-aware instance mining (MIM) algorithm to efficiently mine object instances by combining proposal confidence and instance similarity.
- With the help of memory, we also propose a memory-aware proposal sampling (MPS) algorithm to alleviate the imbalance between positive and negative samples by finding more positive proposals and removing some unreliable negative proposals.

2. Related Work

In this section, we present the two most relevant to this paper, weakly supervised object detection and contrastive learning.

2.1. Weakly Supervised Object Detection

Since Hakan Bilen and Andrea Vedaldi proposed a weakly supervised deep detection network (WSDDN) [20] to combine MIL and CNN into an end-to-end network, most WSOD methods follow [14–19,21,22,26,27] this pipeline to train the weakly supervised detector. In MIL, an image is treated as a bag of proposals. If the image contains an object class, this bag is labelled as positive bag, i.e., at least containing one object instance of this class, otherwise labelled as negative bag. Due to lack of instance-level annotations, MIL is tend to get stuck in a local optimum to locate the most representative part of target objects. Subsequently, most researchers have proposed promising approaches to alleviate this problem. For instance, Kantorov et al. [14] proposed additional and contrastive context-aware guidance models to improve localization by using the surrounding contextual region of proposals. Tang et al. [15] proposed an online instance classifier refinement (OICR) method that uses spatial correlations between proposals to refine mined instances. Wan et al. [21] proposed continuation multiple instance learning (C-MIL) to alleviate the problem that MIL is prone to falling into local optima, which uses some smooth loss function to approximate the original non-convex loss function. Lin et al. [22] proposed object instance mining (OIM) framework to build spatial and appearance graphs of proposals to mine all possible object instances. Furthermore, some methods [16–19] introduce segmentation information to assist instance mining. Shen et al. [16] proposed a recurrent guidance strategy for weakly supervised detection and segmentation, where the detection module generates seeds for semantic segmentation and the segmentation module provides prior information for object detection. Yang et al. [17] proposed an objectness consistent representation method to exploit segmentation map to mine more high-quality proposals. Wei et al. [18] used segmentation context information around proposals to discover tight object bounding boxes. Li et al. [19] leveraged the segmentation map to reweight proposals scores. However, these methods only consider information from a single image, which are difficult to deal with diverse object instances. In this paper, our method explores the semantic correlation beyond the input image to assist object instance mining.

2.2. Contrastive Learning

Contrastive learning [29] has been widely used in unsupervised representation learning (e.g., SimCLR [30] and MoCo [31]), which compares positive and negative pairs to compress together different view representations of the same image and separate view representations of different images. In addition, contrastive learning-based methods [32–35] have also achieved promising performance in other vision tasks. For instance, Yan et al. [33] proposed a semantics-guided contrastive network that introduces contrastive learning into zero-shot object detection to transfer available semantic information for unseen classes. Wu et al. [34] proposed a contrastive learning-based robust object detection algorithm to detect objects under smoky conditions, which applies contrastive learning to maximize the consistency between different augmented views of the same smoke image. Li et al. [35] introduced contrastive learning into remote sensing image semantic segmentation to learn global and local image representations. However, these methods are difficult to directly apply to WSOD that learns the detector based on image-level annotations. In this paper, we introduce contrastive learning into weakly supervised object detection and propose an instance-level contrastive learning framework. To our best knowledge, we are the first to explore contrastive learning for weakly supervised object detection.

3. Method

In this section, we first describe our instance-level contrastive learning (ICL) framework in detail. Then, we present the instance-diverse memory updating (IMU) algorithm, memory-aware instance mining (MIM) algorithm and memory-aware proposal sampling (MPS) algorithm.

3.1. Instance-Level Contrastive Learning

In Figure 2, we present the pipeline of the instance-level contrastive learning (ICL) framework. Given an input image I and the corresponding proposals R generated by the proposal generation methods [36–38], we first extract image features F_I using convolutional neural network. Based on the pre-generated proposals R, we convert image features F_I to proposal features F_R through a RoI-pooling layer, and use two fully connected (FC) Layers to obtain proposal vector representations F_V. By mining reliable instance representations from F_V, we then perform contrastive learning (CL) and store them in the memory bank M. In addition, F_V is fed into several parallel detection heads, where a base head is supervised by the image label $Y = [y_1, y_2, \ldots, y_C]^T \in \mathbb{R}^{C\times 1}$ and K refined heads are supervised by the output results of previous heads, where C is the number of classes. In this paper, we set $K = 3$ to be the same as our baseline method [15].

Unsupervised representation learning [30,31] performs contrastive learning by augmenting image to different views, where views of the same image are pulled closer and views of different images are pulled apart. In this paper, we introduce the contrastive learning of object instance representations to guide the detector to learn the entire representation of the instance. Specifically, we first denote all outputs of refined heads as $(\{\varphi^1, \varphi^2, \ldots, \varphi^K\}, \{t^1, t^2, \ldots, t^K\})$. To mine more reliable instances, we average these outputs to obtain the proposal scores $\varphi = \frac{1}{K}\sum_{k=1}^{K}\varphi^k$ and the bounding box coordinate offsets $t = \frac{1}{K}\sum_{k=1}^{K} t^k$. Applying the coordinate offset t to transform R, we obtain the transformed proposal P. Then, we exploit non-maximum suppression (NMS) to mine as many object instances as possible. For a positive class c, we use NMS to gradually select object instances from the transpose proposal P_c according to the proposal score φ_c from high to low, and remove redundant proposals. Then, we set a score threshold T_1 to obtain more reliable object instances D, and extract the corresponding instance feature representations F_D from F_V. The detailed procedure can be seen in Algorithm 1. For each mined instance representation $q \in F_D$, we utilize the memory instance representation M including all training data to assist each instance learning in current image. Assume that a positive representation from memory bank $k_+ \in M$ represent the same class as q, and a negative representation from memory bank $k_- \in M$ represent different classes. Then, we use the contrastive loss [39] to pull q close to k_+ of the same class while pushing it away from negative keys k_- of other classes, and thus enhance the discrimination and generalization of current instance representation:

$$L_{CL} = -\frac{1}{|D|} \sum_{q \in F_D} \varphi_q log \frac{exp(q \cdot k_+/\tau)}{exp(q \cdot k_+/\tau) + \sum_{k_-} exp(q \cdot k_-/\tau)}, \tag{1}$$

where we take φ_q as the loss weight and τ means the temperature hyperparameter.

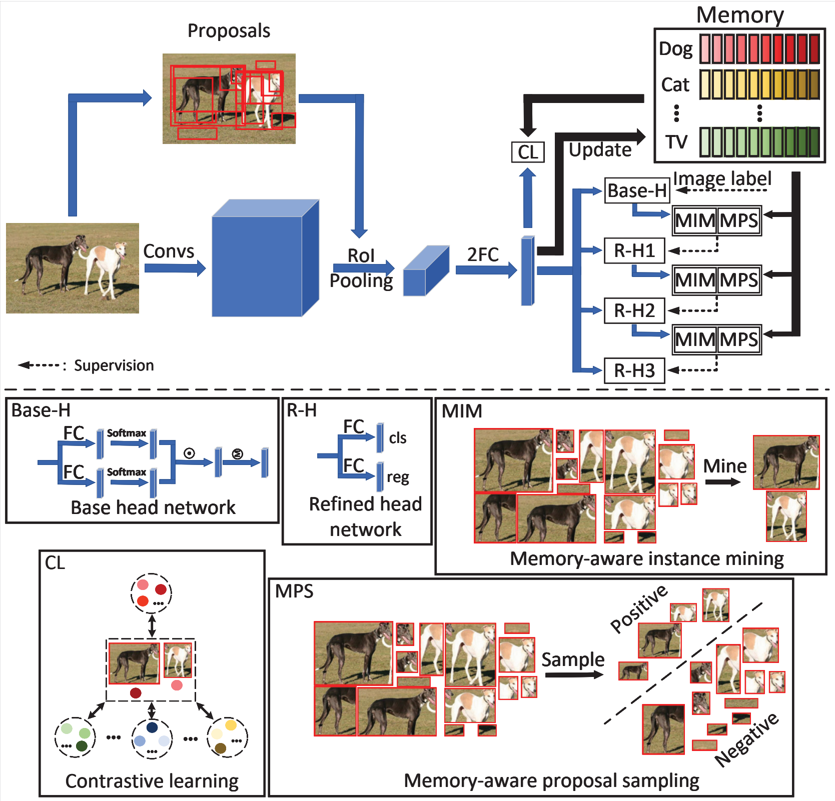


Figure 2. The pipeline of instance-level contrastive learning (ICL) framework. The upper part shows the overall network structure, where only the blue arrows backpropagate the gradients. There are one base head (Base-H) and three refined heads (R-H1, R-H2, R-H3). The base head is supervised by image labels, while each refined head is supervised by the previous parallel head. Dashed arrows indicate the supervision information. The detailed network structure of these heads can be found in the lower boxes, where each box corresponds to a submodule in the pipeline. Three refined heads have the same network structure but do not share parameters. The processes of memory-aware instance mining (MIM) algorithm, memory-aware proposal sampling (MPS) algorithm and contrastive learning (CL) are also shown in boxes.

Algorithm 1 Instance representation mining algorithm.

```
Input: The pre-generated proposals $R$, the pre-defined score threshold $T_1$, the
  image label $Y$, the memory bank $M$, the outputs of instance refined heads
  $(\{\varphi^1, \varphi^2, ..., \varphi^K\}, \{t^1, t^2, \ldots, t^K\})$ and the proposal feature vectors $F_V$.
    (I) average proposal scores $\varphi = \frac{1}{K}\sum_{k=1}^{K}\varphi^k$
    (II) average coordinate offsets $t = \frac{1}{K}\sum_{k=1}^{K}t^k$
    (III) obtain transformed proposals $P$ by adding $t$ to $R$
    (IV) instance representations $F_D = \varnothing$ and the corresponding confidences $\varphi_{F_D} = \varnothing$
    For $c = 0$ to $C + 1$
      If $y_c == 1$ or $c == C + 1$
        (1) $keep = NMS(P_c, \varphi_c)$
        (2) $P_{keep} = P_c[keep]$
        (3) $\varphi_{keep} = \varphi[keep]$
        (4) $D_c = P_{keep}[\varphi_{keep} > T_1]$
        (5) $\varphi_{D_c} = \varphi_{keep}[\varphi_{keep} > T_1]$
        (6) $F_{D_c} = F_V[D_c]$
        (7) $F_D = F_D \cup F_{D_c}$, $\varphi_{F_D} = \varphi_{F_D} \cup \varphi_{D_c}$
Output: $F_D$, $\varphi_{F_D}$.
```

Subsequently, we describe the training of heads. The base head has two parallel branches. One branch uses an FC layer to generate a matrix $x^c \in \mathbb{R}^{C\times|R|}$ ($|R|$ is the number of proposals), which is then input to a class-wise softmax layer: $[\sigma(x^c)]_{ij} = \frac{e^{x^c_{ij}}}{\sum_{q=1}^{C} e^{x^c_{qj}}}$. In another branch, there is an FC layer and a proposal-wise softmax layer to generate another normalized matrix $\sigma(x^r)$, where $x^r \in \mathbb{R}^{C\times|R|}$ and $[\sigma(x^r)]_{ij} = \frac{e^{x^r_{ij}}}{\sum_{q=1}^{|R|} e^{x^r_{iq}}}$. Then, element-wise matrix multiplication is performed on these two matrices to generate proposal scores $x^R = \sigma(x^c) \odot \sigma(x^r)$. Finally, the image class score is calculated by summing all proposal scores: $\phi = \sum_{r=1}^{|R|} x^R$. According to the image label $Y = [y_1, y_2, ..., y_C]^T$, the loss of base head is computed by Equation (2).

$$L_b = -\sum_{c=1}^{C}\{y_c \log\phi_c + (1 - y_c)\log(1 - \phi_c)\}. \tag{2}$$

For K refined heads, their training process is consistent. Specifically, for the k^{th} head, there is a classifier and a regressor. In the classifier, an FC layer and a class-wise softmax are used to generate proposals scores $\varphi^k \in \mathbb{R}^{(C+1)\times|R|}$, where $C + 1$ means background is included. In the regressor, an FC layer is used to produce the coordinate offsets of proposals $t^k \in \mathbb{R}^{4C\times|R|}$, where 4 means the dimension of coordinate offsets (x_1, y_1, x_2, y_2). In order to generate their supervision, we first use the memory-aware instance mining (MIM) algorithm to mine multiple representative object instances B^k based on the score outputs and offset outputs of previous head (φ^{k-1}, t^{k-1}) and memory bank M. The details can be seen in Section 3.3. Then, we use the memory-aware proposal sampling (MPS) algorithm of Section 3.4 to sample negative and positive proposals (R_{pos}, R_{neg}) from proposals R and assign labels for these proposals. For a positive class c, if a proposal p is selected as positive sample, p is labeled as class c, i.e., $y^k_{c,p} = 1$. All negative proposals R_{neg} are labeled as background class $C + 1$. In this way, the classifier can be trained by a cross entropy loss:

$$L^k_{cls} = -\frac{1}{|R_{pos}| + |R_{neg}|}\sum_{p\in R_{pos}\cup R_{neg}}\sum_{c=1}^{C+1}\varphi^k_p y^k_{c,p}\log\varphi^k_{c,p}, \tag{3}$$

where we also use the confidence φ_p^k as the loss weight. For the regressor, only positive proposals R_{pos} are used to calculate loss by the smooth L1 loss [4]:

$$L_{reg}^k = \text{smoothL1}(t^k, T^k), \tag{4}$$

where T^k is the supervision of coordinate offsets.

In summary, our ICL framework can be end-to-end trained in Equation (5).

$$L = L_{CL} + L_b + \sum_{k=1}^{K}(L_{cls}^k + L_{reg}^k). \tag{5}$$

3.2. Instance-Diverse Memory Updating Algorithm

In order to enable the network to memory the instance representations from previous training images, we first initialize $M \in \mathbb{R}^{(C+1)\times N\times L}$, where N means the number of stored instance feature representations in each class and L represents the length of the feature vector F_V. Since instances of the same class differ in size, shape, and appearance, we use multiple feature vectors to store richer instance representations instead of a single vector. We first use the Algorithm 1 to obtain some reliable instance representations F_D from F_V. For each instance representation $f_{d,c} \in F_D$, we calculate the similarity between $f_{d,c}$ and with $M_c = \{f_{c,1}, f_{c,2}, \ldots, f_{c,N}\}$ in Equation (6).

$$S_{d,c} = ||f_{d,c}|| \times ||M_c^{\mathrm{T}}||, \tag{6}$$

where $||\cdot||$, × and T mean L_2 normalization, matrix multiplication and transpose, respectively. Then, we select the most similarity feature $f_{c,j}$ from M_c in Equation (7) to maximize the assistance of the current instance.

$$j = argmax\{S_{d,c}\}, \tag{7}$$

Finally, we update the feature vector $f_{c,j}$ according Equation (8) for the instance contrastive learning of subsequent images.

$$f_{c,j} = r * f_{c,j} + (1-r) * \varphi_{f_{d,c}} * f_{d,c}, \tag{8}$$

where r is the momentum coefficient [31] and $\varphi_{f_{d,c}}$ is the confidence of instance representation, which aim to control the weight balance between previous instance representation and current representation. The whole process can be seen in the Algorithm 2.

Algorithm 2 Instance-diverse memory updating algorithm.

```
Input: The pre-generated proposals $R$, the pre-defined score threshold $T_1$, the
  image label $Y$, the memory bank $M$, the outputs of instance refined heads
  $(\{\varphi^1, \varphi^2, \ldots, \varphi^K\}, \{t^1, t^2, \ldots, t^K\})$ and the proposal feature vectors $F_V$.
    (I) obtain reliable instance representations $F_D$ using Algorithm 1
    For each representation $f_{d,c}$ in $F_D$
      (a) compute the similarity between $f_{d,c}$ and with $M_c$ in Equation (6)
      (b) choose the most similarity feature $f_{c,j}$ from $M_c$ in Equation (7)
      (c) update $f_{c,j}$ in Equation (8)
Output: Updated memory $M$.
```

3.3. Memory-Aware Instance Mining Algorithm

With the help of memory bank M, we propose a memory-aware instance mining (MIM) algorithm to effectively mine some reliable object instances. Different from our baseline [15], which only selects the top-scoring proposal as pseudo instance annotations, we comprehensively consider the confidence of proposals and the similarity between proposal features and memory bank covering previous training data to effectively mine object instances. Specifically, we first calculate the similarity S between F_V and M according to Equation (9).

$$S = ||F_V|| \times ||M^{\mathrm{T}}||. \tag{9}$$

Then, we select the highest similarity along the N feature vectors and apply the class-wise softmax to generate memory-base confidence φ_M through Equation 10.

$$\varphi_M = softmax(\max_N\{S\}). \tag{10}$$

For the k^{th} branch in instance refinement heads, we further calculate the combination confidence ψ^k in Equation (11).

$$\psi^k = \varphi^k + \mu\varphi_M, \tag{11}$$

where μ is the combination coefficient. Next, we use the NMS algorithm to remove redundant proposals and set a score threshold T_2 to remove unreliable proposals. In this way, we can obtain some reliable instances B^k. More details can be found in Algorithm 3.

Algorithm 3 Memory-aware instance mining algorithm.

```
Input: The pre-generated proposals $R$, the pre-defined score threshold $T_2$, the image label
  $Y$, the memory bank $M$, the outputs of $k^{th}$ instance refined head $(\varphi^k, t^k)$ and the proposal
  feature vectors $F_V$.
    (I) obtain transformed proposals $R_{t^k}$ by adding $t^k$ to $R$
    (II) calculate the memory-based confidence $\varphi_M$ by Equation (10)
    (III) compute the combination confidence $\psi^k$ with Equation (11)
    For $c = 0$ to $C$
      If $y_c == 1$
        (1) $keep = NMS(R_{t^k}, \psi_c^k)$
        (2) $R_1 = R_{t^k}[keep]$
        (3) $\psi_1 = \psi_c^k[keep]$
        (4) $B^k = R_1[\varphi_1 > T_2]$
        (5) $\psi_{B^k}^k = \varphi_1[\varphi_1 > T_2]$
Output: $B^k$, $\psi_{B^k}^k$.
```

3.4. Memory-Aware Proposal Sampling Algorithm

After mining object instances, we further propose a memory-aware proposal sampling (MPS) algorithm to effectively sample positive and negative proposals from R. Some methods [15–19,26] simply divide R into two parts by computing the IoU with B^k: highly overlapped proposals are taken as positive samples, and the rest are taken as negative samples, while ignoring the imbalance of positive and negative samples with overwhelmingly negative proposals. To alleviate this problem, we leverage the memory bank to select more positive proposals and remove some unreliable negative proposals. We first calculate the IoU between B^k and R to separate R into two parts R_1^k and R_2^k in Equation (12).

$$\begin{cases} R_1^k = \left\{p \in R | \exists b \in B^k, IoU(p,b) \geqslant 0.5\right\} \\ R_2^k = \left\{p \in R | \forall b \in B^k, 0.1 < IoU(p,b) < 0.5\right\} \end{cases} \tag{12}$$

Then, we extract the feature representations $F_{R_2^k}$ of R_2^k from F_V. For each positive class c, we compute the similarity between $F_{R_2^k}$ and M_c, and use Equation (13) to choose the most similar proposal p_c into R_1.

$$j = argmax(|F_{R_2^k}| \times |M_c^{\mathrm{T}}|). \tag{13}$$

For the remaining proposals in R_2^k, we compute the similarity $S_{R_2^k} = \underset{N}{max}\{|F_{R_2^k}| \times |M_c^{\mathrm{T}}|\}$ between R_2^k and background information M_{C+1} and sort $S_{R_2^k}$ according to similarity from high to low. Finally, We removed the last low-similarity $1/\lambda$ proposals from R_2^k to obtain the negative samples. More details can be found in Algorithm 4.

Algorithm 4 Memory-aware proposal sampling algorithm.

```
Input: The pre-generated proposals $R$, the image label $Y$, the memory bank $M$, the mined
  object instance $(B^k, \psi^k_{B^k})$ and the proposal feature vectors $F_V$.
    (I) positive samples $R_{pos} = \varnothing$, negative samples $R_{neg} = \varnothing$
    (II) calculate $IoU(B^k, R)$
    (III) separate $R$ into two parts $R_1^k$ and $R_2^k$ using Equation (12).
    (IV) $R_{pos} = R_{pos} \cup R_1^k$
    (V) extract feature representations $F_{R_2^k}$ from $F_V$
    For $c = 0$ to $C$
      If $y_c == 1$
        (1) calculate the similarity between $F_{R_2^k}$ and $M_c$
        (2) select the most similar proposal $p_c$ from $R_2^k$ by Equation (13)
        (3) $R_{pos} = R_{pos} \cup p_c$, $R_2^k = R_2^k / p_c$, $F_{R_2^k} = F_{R_2^k} / F_{p_c}$
    (VI) $S_{R_2^k} = \underset{N}{max}\{|F_{R_2^k}| \times |M_c^{\mathrm{T}}|\}$
    (VII) sort $S_{R_2^k}$ from from high to low
    (VIII) obtain $R_{neg}$ by removing the last low-similarity $1/\lambda$ proposals from $R_2^k$
Output: $R_{pos}$, $R_{neg}$.
```

3.5. Test

After training, only the instance refined heads are used for testing. We perform the same operations as our baseline method [15]. We average the outputs of all refined heads to generate the final detection results.

4. Experiments

In this section, we first introduce experimental data and evaluation criteria, and elaborate on experimental details. Then we validate the advantages of our method by comparing with some recent methods. Finally, we conduct extensive ablation experiments to demonstrate the effectiveness of our method.

4.1. Datasets and Evaluation Measures

We conduct experiments on PASCAL VOC2007 [40] and VOC2012 [41] datasets, which are widely used in weakly supervised object detection setting [14–22,25–27]. In the PASCAL VOC2007 dataset, there are 9962 images belonging to 20 categories. These images are divided into three sets: *train*, *val*, *test*. According to the widely used WSOD setting, the *trainval* set (5011 images) is used for training. The PASCAL VOC2012 dataset has 22531 images split into *train*, *val* and *test* sets. The *trainval* set has 11540 images for training. It is important to note that all experiments have only image-level labels for training. For evaluation, there are two evaluation measures mean average precision (mAP [40]) and correct localization (CorLoc [42]). mAP is the standard PASCAL VOC protocol, which first computes the average precision (AP) for each class and then averages over all classes. AP for each class is obtained by calculating the area under the precision-recall curve. The mAP is used to evaluate performance on the *test* set. The second metric CorLoc is used to measure the localization accuracy of the *trainval* set. For each class, CorLoc is calculated as the ratio of images where at least one object is correctly localized. Both mAP and CorLoc are based on the PASCAL criterion. The object is considered to be successfully detected, when the intersection over union (IoU) between the ground-truth and predicted boxes is greater than 0.5.

4.2. Experimental Details

All experiments are performed on the Detectron2 (https://github.com/facebookresearch/detectron2) deep learning framework and 4 NVIDIA GTX 1080ti GPUs. Following our baseline method OICR [15], we use the the VGG16 [2] model as our backbone, pre-trained on the ImageNet dataset [43]. For pre-generated proposals, we use multiscale combinatorial grouping (MCG) method [38] to generate approximately 2000 proposals per image. During the training phase, we set the learning rate to 0.001 for the first 28 epochs and divide it by 10 for the next 12 epochs. In addition, we set the momentum and weight decay to 0.9 and 0.0005, respectively. The mini-batch size is set to 4, i.e., an image is run by one GPU. Regarding the data augmentation, we use 5 scales {480, 576, 688, 864, 1200} to randomly resize the shortest side of the image and make the longest side no more than 2000, where the random horizontal flips are also used. During the test stage, We average the output of all augmented data to generate final detection results. For Hyperparameters in our method, we set $K = 3$, $N = 5$, $\mu = 0.1$, $1/\lambda = 1/4$, and $T_1 = T_2 = 0.5$. All settings in the PASCAL VOC2007 and VOC2012 datasets are the same.

4.3. Comparison with Other Methods

On PASCAL VOC2007 and VOC2012 datasets, we compare our method with some recent methods [14–22,25–27] to present our advantages.

For the PASCAL VOC2007 dataset, we present the detection performance (mAP) and localization accuracy (CorLoc) in Tables 1 and 2, respectively. In terms of mAP, our method ICL achieves a detection performance of 55.4%, which brings a significant improvement (about 14.2%) compared to our baseline OICR [15] (41.2%). Our method also outperforms methods [16–19] that exploit segmentation information to learn instance representation. For example, compare with [17], our method has an advantage of about 4.8%. Furthermore, our method also has some improvements (about 1.9%) compared to recent methods SLV[27] and D-MIL [25]. In terms of CorLoc, our method ICL achieves 74.0% localization accuracy. Compared with our baseline (60.6%), our method improves the performance by about 13.4%. Compared to the segmentation-assisted method WS-JDS [16] or SDCN [19], our method improves the performance by more than 7.2%. In addition, our method also shows significant advantages (more than 3%) compared to recent methods SLV [27] and D-MIL [25].

Table 1. Comparison with other methods on Pascal VOC2007 *test* set. ★ means our baseline.

Method	aer	bik	bir	boa	bot	bus	car	cat	cha	cow	tab	dog	hor	mot	per	pla	she	sof	tra	tv	mAP
WSDDN [20]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
Kantorov et al. [14]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
OICR [15] ★	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
PCL [26]	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63.0	43.5
C-MIL [21]	62.5	58.4	49.5	32.1	19.8	70.5	66.1	63.4	20.0	60.5	52.9	53.5	57.4	68.9	8.4	24.6	51.8	58.7	66.7	63.5	50.5
Lin et al. [22]	55.6	67.0	45.8	27.9	21.1	69.0	68.3	70.5	21.3	60.2	40.3	54.5	56.5	70.1	12.5	25.0	52.9	55.2	65.0	63.7	50.1
TS^{2}C [18]	59.3	57.5	43.7	27.3	13.5	63.9	61.7	59.9	24.1	46.9	36.7	45.6	39.9	62.6	10.3	23.6	41.7	52.4	58.7	56.6	44.3
WS-JDS [16]	52.0	64.5	45.5	26.7	27.9	60.5	47.8	59.7	13.0	50.4	46.4	56.3	49.6	60.7	25.4	28.2	50.0	51.4	66.5	29.7	45.6
SDCN [19]	59.8	67.1	32.0	34.7	22.8	67.1	63.8	67.9	22.5	48.9	47.8	60.5	51.7	65.2	11.8	20.6	42.1	54.7	60.8	64.3	48.3
Yang et al. [17]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.6
SLV [27]	65.6	71.4	49.0	37.1	24.6	69.6	70.3	70.6	30.8	63.1	36.0	61.4	65.3	68.4	12.4	29.9	52.4	60.0	67.6	64.5	53.5
D-MIL [25]	60.4	71.3	51.1	25.4	23.8	70.4	70.3	71.9	25.2	63.4	42.6	67.1	57.7	70.1	15.5	26.6	58.7	63.3	66.9	67.6	53.5
ICL	61.9	73.0	44.0	33.3	32.9	75.3	74.7	73.8	2.6	70.6	62.0	60.8	72.2	71.3	26.0	25.4	57.3	57.7	72.7	60.9	**55.4**

For the PASCAL VOC2012 dataset, we show both detection performance (mAP) and localization accuracy (CorLoc) in the Table 3. Our method achieves 50.1% mAP and 70.4% CorLoc, which are 12.2% and 8.3% improvement over the baseline, respectively. Compare to some recent methods [25,27], our method also bring some gains. In terms of mAP, our method outperforms the methods [27] and [25] by 0.9% and 0.5%, respectively. In terms of CorLoc, our method brings gains of 1.2% and 0.3%, respectively. These results further demonstrate the effectiveness of our method.

Table 2. Comparison with other methods on Pascal VOC2007 *trainval* set. ★ means our baseline.

Method	aer	bik	bir	boa	bot	bus	car	cat	cha	cow	tab	dog	hor	mot	per	pla	she	sof	tra	tv	mean
WSDDN [20]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
Kantorov et al. [14]	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
OICR [15] ★	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
PCL [26]	79.6	85.5	62.2	47.9	37.0	83.8	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	68.5	75.7	78.9	62.7
C-MIL [21]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
Lin et al. [22]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.2
TS^{2}C [18]	84.2	74.1	61.3	52.1	32.1	76.7	82.9	66.6	42.3	70.6	39.5	57.0	61.2	88.4	9.3	54.6	72.2	60.0	65.0	70.3	61.0
WS-JDS [16]	82.9	74.0	73.4	47.1	60.9	80.4	77.5	78.8	18.6	70.0	56.7	67.0	64.5	84.0	47.0	50.1	71.9	57.6	83.3	43.5	64.5
SDCN [19]	85.8	83.1	56.2	58.5	44.7	80.2	85.0	77.9	29.6	78.8	53.6	74.2	73.1	88.4	18.2	57.5	74.2	60.8	76.1	79.2	66.8
SLV [27]	84.6	84.3	73.3	58.5	49.2	80.2	87.0	79.4	46.8	83.6	41.8	79.3	88.8	90.4	19.5	59.7	79.4	67.7	82.9	83.2	71.0
D-MIL [25]	81.3	82.0	72.7	48.9	42.0	80.2	86.1	78.5	43.9	80.2	42.2	76.5	68.7	91.2	32.7	56.0	81.4	69.6	78.7	79.9	68.7
ICL	85.3	88.9	65.5	57.5	57.4	86.0	90.7	85.8	15.1	88.7	78.0	74.4	89.2	93.5	39.2	57.6	88.5	71.6	86.2	80.1	**74.0**

Table 3. Comparison with other methods on Pascal VOC 2012 dataset. ★ means our baseline.

Method	mAP	CorLoc
Kantorov et al. [14]	35.3	54.8
OICR [15] ★	37.9	62.1
PCL [26]	40.6	63.2
C-MIL [21]	46.7	67.4
Lin et al. [22]	45.3	67.1
TS^{2}C [18]	40.0	64.4
WS-JDS [16]	39.1	63.5
SDCN [19]	43.5	67.9
SLV [27]	49.2	69.2
D-MIL [25]	49.6	70.1
ICL	**50.1**	**70.4**

4.4. Ablation Study

In this part, we conduct extensive experiments to further discuss the effects of main components of our method. Without loss of generality, all experiments are performed on the PASCAL VOC2007 dataset.

The effect of IMU algorithm. We first analyze the effect of IMU algorithm on our method ILC. In Table 4, after removing IMU, our method achieves 53.6% mAP and 72.9% CorLoc, respectively. There are 1.8% performance reduction and 1.1% accuracy reduction in terms of mAP and CorLoc, respectively, which proves the effectiveness of the instance-diverse memory updating algorithm. Furthermore, we analyze the effect of the number of feature vector N on the IMU algorithm in Figure 3. We can see that both mAP and CorLoc first increase and then decrease as N increases. When N is too small, the memory bank is difficult to store the diversity of instance representations well, and when N is too large, it is easy to cause there are internal differences during the learning of instance representations. In this paper, we recommend setting $N = 5$ to balance the number of stored instance vectors.

Table 4. The contribution of each component of our method. Both PASCAL metrics and COCO metrics are applied.

Method	PASCAL Metrics		COCO Metrics					
	mAP	CorLoc	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Baseline	41.2	60.6	14.9	37.0	10.7	1.8	8.6	19.4
ICL	**55.4**	**74.0**	**20.8**	**48.8**	**14.4**	2.3	10.3	**27.1**
ICL w/o IMU	53.6	72.9	19.8	46.8	14.2	2.8	10.1	25.9
ICL w/o MIM	49.7	69.6	18.0	43.1	12.5	2.1	8.9	23.8
ICL w/o MPS	52.9	71.4	19.5	46.7	13.5	**3.3**	**10.5**	25.0

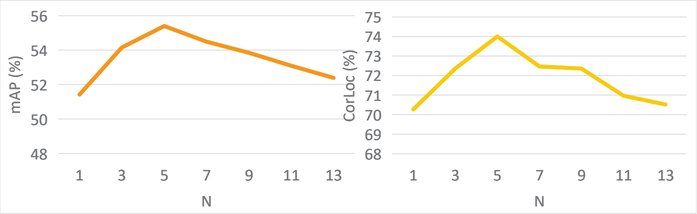


Figure 3. The effect of number of vectors N.

The effect of MIM algorithm. As shown in the Table 4, MIM brings 5.7% and 4.4% gains to ICL in mAP and CorLoc, which shows the effectiveness of memory-aware instance mining algorithm (MIM). In addition, we also analyze the effect of memory on MIM by setting different combination coefficients μ in Figure 4. During the change of μ from 0 to 1, we can see that both the detection performance and the localization accuracy are the highest at $\mu = 0.1$, which demonstrates that it is useful to introducing the similarity between memory features of previous training data and proposal features for mining effective instances. When μ becomes larger, the memory from previous images may hinder the learning of new instances from the current image, resulting in performance degradation. Therefore, we set $\mu = 0.1$ in this paper.

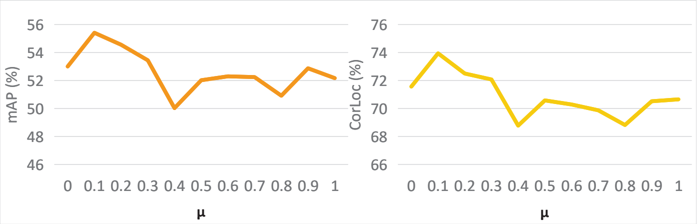


Figure 4. The effect of combination coefficient μ.

The effect of MPS algorithm. Removing MPS from ICL, our method achieves 52.9% detection performance (mAP) and 71.4% localization accuracy (CorLoc). There are 2.5% and 2.6% reductions in mAP and CorLoc, respectively, which shows the effectiveness of the memory-aware proposal sampling algorithm. In addition, we analyze the effect of the removal coefficient $1/\lambda$ on MPS in Table 5. We achieve the best performance when $1/\lambda = 1/4$. Continuing to increase $1/\lambda$ may remove too many negative samples and affect the training of the detector. When $1/\lambda$ is too small, it cannot achieve the purpose of balancing positive and negative samples.

Table 5. The effect of the removal coefficient $1/\lambda$.

$1/\lambda$	mAP	CorLoc
1/10	52.7	70.8
1/8	52.0	70.1
1/6	53.6	72.6
1/4	**55.4**	**74.0**
1/2	52.4	70.8

The performance of COCO metrics. In Table 4, we also analyze the contribution of each component under the COCO metrics [44]. The performance of each component on AP, AP_{50}, and AP_{75} is similar to that under the PASCAL metrics. For AP_S, AP_M and AP_L, the objects are divided into three sizes of small, medium and large for evaluation. Our method ICL can achieve the best performance on large objects, while removing MPS algorithm can achieve better performance on small and medium objects. Compared with small and medium-sized objects that are difficult to perceive, MPS algorithm is more conducive to sampling region proposals of large objects.

The analysis of training process. In Figure 5, we further provide training loss curves to verify the rationality of our method. We can see that the loss curves of refined heads rise first and then decrease to convergence. The rising phase of the loss is due to the weight, which is the confidence of mined object instances. At the beginning of training, the low discrimination of the model makes the confidence of object instances very low (almost close to 0). As model capabilities increase, the confidence starts to increase and so does the loss. Since confidence range is $[0, 1]$, the loss will reach the maximum value, and finally start to decrease due to the enhanced model generalization until it converges. In Figure 6, we also provide the performance of the model during training. Both mAP and CorLoc continue to increase, which further demonstrates the effectiveness of our method.

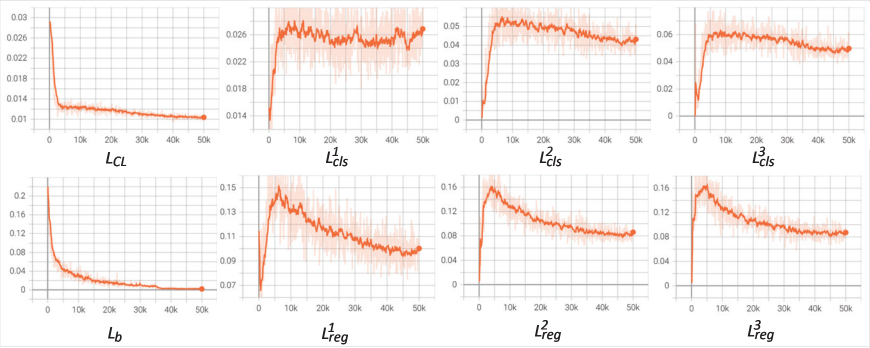


Figure 5. Training loss.

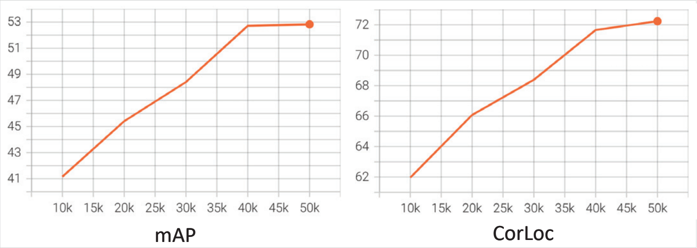


Figure 6. Model performance during the training process. Evaluations are performed every 10K without data augmentation.

Qualitative results. In Figure 7, we provide qualitative results to more intuitively compare the proposed ICL with our baseline. On the *trainval* set of PASCAL VOC2007 dataset, we compare the learned object instances in Figure 7a. For the simple image in the first column, the baseline method can learn effective information about the car well. For the horse in the second column and the cow in the third column, when the foreground and background are relatively similar or the objects are occluded, the instances learned by the baseline method may contain more background. Our method ICL can learn more reliable object instances guided by instance correlations. On the *test* set, we compare the detection results in Figure 7b. Since the baseline method is more easily disturbed by background information during the training process, its detection results also contain more background information, such as the first two columns. Our method can better locate the boundary of the object. When there is interaction between objects, such as the third column, our method can also provide better detection results. For a more comprehensive analysis of our method, we present failure cases in the last column. For example, for the smaller aeroplane in

Figure 7a, it is difficult for our method to learn its instance representation. Our method also fails to detect the highly overlapping sheep and distant little sheep in Figure 7b.

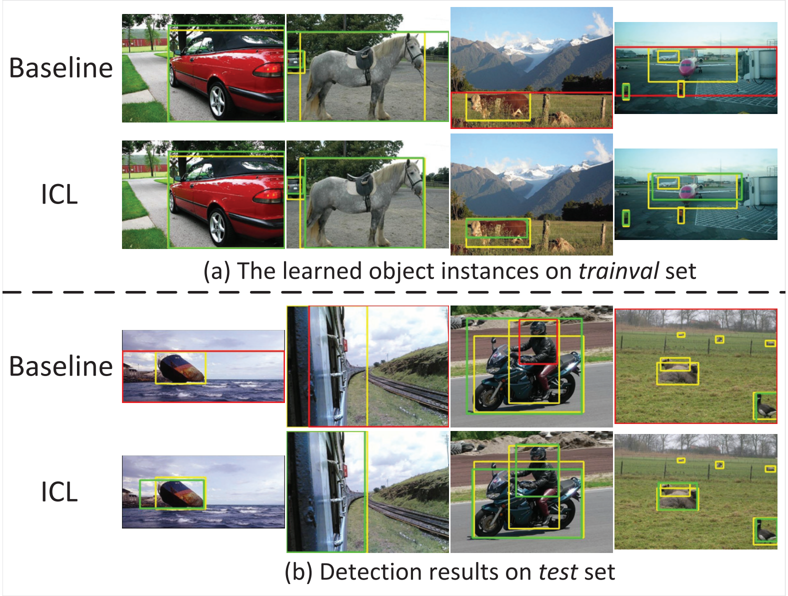

Baseline
ICL
(a) The learned object instances on *trainval* set
Baseline
ICL
(b) Detection results on *test* set

Figure 7. Qualitative results on PASCAL VOC2007 dataset. Yellow boxes mean ground truths. Green and red boxes represent correct and failed cases, respectively.

5. Conclusions

In this paper, we propose an instance-level contrastive learning (ICL) framework to guide the weakly supervised detector to learning entire instance representations by constructing instance correlations with other images. To store diverse object instance representations in a memory bank, we propose an instance-diverse memory updating (IMU) algorithm. With the help of memory, we further propose a memory-aware instance mining (MIM) algorithm to effectively mine object instances. To alleviate the imbalance of positive and negative proposals, we propose a memory-aware proposal sampling (MPS) algorithm. We conduct extensive experiments on PASCAL VOC2007 and VOC2012 datasets to verify the effectiveness of our method.

Our proposed method mines object instance representations from other images and stores them in a memory bank to guide instance learning on the current image. If the memory contains noisy representations, it will make the learned object instances inaccurate. The performance of weakly supervised detectors is also limited by the quality of the stored representations. In order to mine more reliable instance representations, our future studies will explore contextual information of region proposals or segmentation information of images to perceive object boundaries and locate object instances accurately.

Author Contributions: Conceptualization, M.Z.; methodology, M.Z.; software, M.Z.; validation, M.Z.; formal analysis, M.Z.; investigation, M.Z.; resources, M.Z.; data curation, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, B.Z.; visualization, M.Z.; supervision, B.Z.; project administration, M.Z.; funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China under Grant 61720106004, and in part by the 111 Projects under Grant B17008.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The PASCAL VOC2007 and VOC2012 datasets can be available in the following link: http://host.robots.ox.ac.uk/pascal/VOC/.

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, Spain, 3–8 December 2012; pp. 1097–1105.
2. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
3. 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.
4. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
9. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 9627–9636.
10. Qiu, H.; Li, H.; Wu, Q.; Shi, H. Offset bin classification network for accurate object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 13188–13197.
11. Zhao, H.; Zhou, Y.; Zhang, L.; Peng, Y.; Hu, X.; Peng, H.; Cai, X. Mixed YOLOv3-LITE: A lightweight real-time object detection method. *Sensors* **2020**, *20*, 1861. [CrossRef] [PubMed]
12. Lian, J.; Yin, Y.; Li, L.; Wang, Z.; Zhou, Y. Small object detection in traffic scenes based on attention feature fusion. *Sensors* **2021**, *21*, 3031. [CrossRef] [PubMed]
13. Xiang, Y.; Zhao, B.; Zhao, K.; Wu, L.; Wang, X. Improved Dual Attention for Anchor-Free Object Detection. *Sensors* **2022**, *22*, 4971. [CrossRef] [PubMed]
14. Kantorov, V.; Oquab, M.; Cho, M.; Laptev, I. Contextlocnet: Context-aware deep network models for weakly supervised localization. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 350–365.
15. Tang, P.; Wang, X.; Bai, X.; Liu, W. Multiple instance detection network with online instance classifier refinement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 2843–2851.
16. Shen, Y.; Ji, R.; Wang, Y.; Wu, Y.; Cao, L. Cyclic guidance for weakly supervised joint detection and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 697–707.
17. Yang, K.; Zhang, P.; Qiao, P.; Wang, Z.; Dai, H.; Shen, T.; Dou, Y. Rethinking segmentation guidance for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 946–947.
18. Wei, Y.; Shen, Z.; Cheng, B.; Shi, H.; Xiong, J.; Feng, J.; Huang, T. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In Proceedings of the European Conference on ComputerbVision, Salt Lake City, UT, USA, 18–23 June 2018; pp. 434–450.
19. Li, X.; Kan, M.; Shan, S.; Chen, X. Weakly supervised object detection with segmentation collaboration. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 9735–9744.
20. Bilen, H.; Vedaldi, A. Weakly supervised deep detection networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2846–2854.
21. Wan, F.; Liu, C.; Ke, W.; Ji, X.; Jiao, J.; Ye, Q. C-mil: Continuation multiple instance learning for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2199–2208.
22. Lin, C.; Wang, S.; Xu, D.; Lu, Y.; Zhang, W. Object instance mining for weakly supervised object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11482–11489.
23. Xu, Y.; Zhou, C.; Yu, X.; Xiao, B.; Yang, Y. Pyramidal multiple instance detection network with mask guided self-correction for weakly supervised object detection. *IEEE Trans. Image Process.* **2021**, *30*, 3029–3040. [CrossRef] [PubMed]
24. Wu, Z.; Wen, J.; Xu, Y.; Yang, J.; Li, X.; Zhang, D. Enhanced Spatial Feature Learning for Weakly Supervised Object Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. [CrossRef] [PubMed]

25. Gao, W.; Wan, F.; Yue, J.; Xu, S.; Ye, Q. Discrepant multiple instance learning for weakly supervised object detection. *Pattern Recognit.* **2022**, *122*, 108233. [CrossRef]
26. Tang, P.; Wang, X.; Bai, S.; Shen, W.; Bai, X.; Liu, W.; Yuille, A. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 176–191. [CrossRef] [PubMed]
27. Chen, Z.; Fu, Z.; Jiang, R.; Chen, Y.; Hua, X.S. Slv: Spatial likelihood voting for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 12995–13004.
28. Dietterich, T.G.; Lathrop, R.H.; Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **1997**, *89*, 31–71. [CrossRef]
29. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 1735–1742.
30. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 1597–1607.
31. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 9729–9738.
32. Sun, B.; Li, B.; Cai, S.; Yuan, Y.; Zhang, C. Fsce: Few-shot object detection via contrastive proposal encoding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual Event, 19–25 June 2021; pp. 7352–7362.
33. Yan, C.; Chang, X.; Luo, M.; Liu, H.; Zhang, X.; Zheng, Q. Semantics-guided contrastive network for zero-shot object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef]
34. Wu, W.; Chang, H.; Zheng, Y.; Li, Z.; Chen, Z.; Zhang, Z. Contrastive Learning-Based Robust Object Detection Under Smoky Conditions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LO, USA, 21–24 June 2022; pp. 4295–4302.
35. Li, H.; Li, Y.; Zhang, G.; Liu, R.; Huang, H.; Zhu, Q.; Tao, C. Global and local contrastive self-supervised learning for semantic segmentation of HR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5618014. [CrossRef]
36. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
37. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
38. Arbeláez, P.; Pont-Tuset, J.; Barron, J.T.; Marques, F.; Malik, J. Multiscale combinatorial grouping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 328–335.
39. Oord, A. V. D.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
40. 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]
41. Everingham, M.; Eslami, S.M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
42. Deselaers, T.; Alexe, B.; Ferrari, V. Weakly supervised localization and learning with generic knowledge. *Int. J. Comput. Vis.* **2012**, *100*, 275–293. [CrossRef]
43. 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 IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009; pp. 248–255.
44. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.

sensors

Article

Deep Non-Line-of-Sight Imaging Using Echolocation

Seungwoo Jang [1], Ui-Hyeon Shin [1] and Kwangsu Kim [2,*]

[1] Department of Artificial Intelligence, Sungkyunkwan University, Suwon 16419, Korea
[2] College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, Korea
* Correspondence: kim.kwangsu@skku.edu

Abstract: Non-line-of-sight (NLOS) imaging is aimed at visualizing hidden scenes from an observer's (e.g., camera) viewpoint. Typically, hidden scenes are reconstructed using diffused signals that emit light sources using optical equipment and are reflected multiple times. Optical systems are commonly adopted in NLOS imaging because lasers can transport energy and focus light over long distances without loss. In contrast, we propose NLOS imaging using acoustic equipment inspired by echolocation. Existing acoustic NLOS is a computational method motivated by seismic imaging that analyzes the geometry of underground structures. However, this physical method is susceptible to noise and requires a clear signal, resulting in long data acquisition times. Therefore, we reduced the scan time by modifying the echoes to be collected simultaneously rather than sequentially. Then, we propose end-to-end deep-learning models to overcome the challenges of echoes interfering with each other. We designed three distinctive architectures: an encoder that extracts features by dividing multi-channel echoes into groups and merging them hierarchically, a generator that constructs an image of the hidden object, and a discriminator that compares the generated image with the ground-truth image. The proposed model successfully reconstructed the outline of the hidden objects.

Keywords: non-line-of-sight; acoustic sensing; depth estimation; deep learning

Citation: Jang, S.; Shin, U.-H.; Kim, K. Deep Non-Line-of-Sight Imaging Using Echolocation. *Sensors* **2022**, *22*, 8477. https://doi.org/10.3390/s22218477

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 13 October 2022
Accepted: 31 October 2022
Published: 3 November 2022

1. Introduction

Over the past few decades, various methods have been proposed to reconstruct hidden scenes that are not visible from an observer's (e.g., camera) perspective. A prominent approach is to use optical equipment to emit a light source towards a relay wall and analyze the scattered light reflected from the hidden scene. Optical equipment primarily consists of a light source (e.g., pulsed laser, continuous laser, and modulated laser) and a detector (e.g., striped camera, single-photon avalanche diode (SPAD), time-of-flight (ToF) camera, and conventional camera). The visible light used by laser tends not to be diffracted owing to its short wavelength. Non-diffractive visible light requires a one-to-one scan. Therefore, more time is required if the size of the hidden scenes is enormous. These physical limitations of visible light, incurring a long time to scan, make it challenging to apply NLOS imaging in real-world scenarios.

Due to these data collection difficulties, applying deep learning models in NLOS imaging is still at an early stage. In recent years, many studies have used synthesized data to overcome the absence of datasets. Chen et al. [1] synthesized the data by rendering diffuse three-bounce of a steady-state laser. Then, they used a U-Net structure to reconstruct the hidden scene. Chopite et al. [2] generated a time histogram image of the photon and used it for training. After that, they extracted features using a 3D → 2D encoder structure and decoded them by upsampling. Tancik et al. [3] performed object positioning and recognition with training data that rendered the flash's reflections. In addition to the above studies, most deep learning models use synthetic data centered on optical systems. However, the synthesized data does not consider noise generated in real-world scenarios, so the generalization performance is poor. To close this gap, we designed NLOS imaging using acoustics and collected real-world datasets without soundproofing.

In contrast to visible light, sound has a long wavelength and diffracts well, which enables the perception of hidden objects. In nature, certain animals (e.g., bats, dolphins, oilbirds, and swiftlets) have highly specialized auditory abilities [4] that enable them to see beyond walls, which is called echolocation. They emit sound and perceive echoes reflected from numerous objects in the environment. Subsequently, they obtain spatial information by analyzing the relative intensity and arrival time delays in the received echoes.

Animals that use echolocation recognize a target's direction, distance, size, and shape based on spatial information. Furthermore, they can even perceive NLOS areas behind obstacles. Recently, methods to reconstruct images or depth maps from echoes based on the innate sound mechanisms of animals have been proposed. Batvision [5] used only echoes for image reconstruction and depth map estimation. Gao et al. [6], and Parida et al. [7] demonstrated that depth estimation is more accurate when used with echoes compared to when using RGB image alone.

Lindell et al. [8] proposed acoustic NLOS (Figure 1) as a computational approach to calculate the time-of-flight (ToF) of received echoes. They collected data with a single input, multiple outputs (SIMO), wherein a single sound emitted sequentially from an array of speakers and recorded from an array of microphones. The hidden object is reconstructed by calculating the ToF of the recorded echoes in each array. Conventional physical models require precise signals because they are susceptible to noise. Therefore, they required several minutes to scan the hidden scenes.

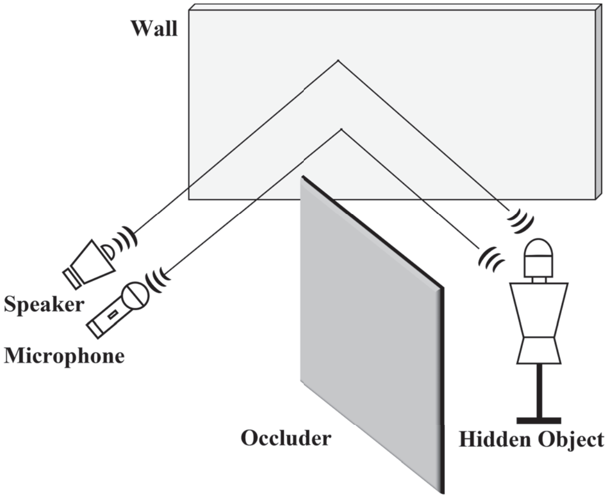


Figure 1. Typical acoustic NLOS setup. Object hidden by the obstacles is not directly visible from the observer's point of view. The speakers emitted chirp signals toward the relay wall, and the microphones recorded the signals reflected off the hidden object.

To reduce the data collection time, we used the MIMO (multiple input, multiple outputs) methods that differ from the previous data collection method, such as SIMO (single input, multiple outputs), as shown in Figure 2. Although the MIMO method reduced the data collection time, using the existing physical model is not easy because the signals interfere with each other. Therefore, we propose an end-to-end deep learning approach that directly learns depth map representation from echoes.

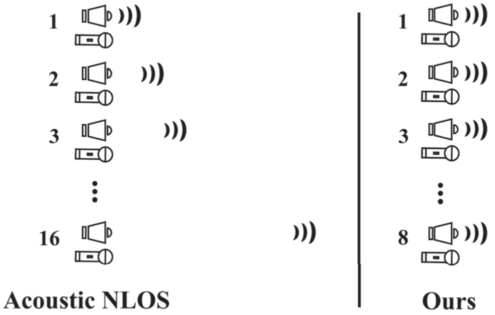


Figure 2. Comparison of the data collecting methods used by our acoustic system and the acoustic NLOS.

The acoustic system we designed collects 64 echoes from different locations. Unlike a general feature extraction method that uses all the channels as input, we designed an encoder for multichannel data to preserve the spatial information of each echo. Then, we split the channels of the echoes and passed them through each encoder to extract the hidden object features. The feature extracted from the encoder is a hierarchical structure that merged with an adjacent channel and entered the encoder again. In this way, we effectively extract features.

The major contributions of this work are as follows:

- We propose an end-to-end deep-learning model that reconstructs a depth map from echoes.
- We designed a hierarchical feature extractor for acoustic NLOS imaging.
- To our knowledge, this is the first work that uses a deep-learning model for acoustic NLOS imaging.

2. Related Work

2.1. Non-Line-of-Sight Image Reconstruction

NLOS imaging is a method for reconstructing hidden scenes that are not visible from the observer's point of view. This reconstruction of a hidden object is achieved by analyzing diffused signals that are reflected several times. The diffused signal returns through the relay wall-hidden object-relay wall. As the signal is reflected three times, it is challenging for high-quality reconstruction because of the low SNR (signal-to-noise ratio). Moreover, it is easy to be exposed to various noises. Methods that can increase the SNR include improving hardware or reconstruction algorithms. In this paper, we focus on the reconstruction algorithm.

Reconstructing hidden objects using signals can be divided into physical and deep-learning models. Conventional physical approaches are promising for NLOS image reconstruction because the diverse hardware, setup, and environment make it challenging to produce large-scale datasets. Consequently, most deep-learning approaches use synthesized data. By contrast, we collected real data because the synthesized data's constraints differ from those collected in the real world.

The physical model is reconstructed by inversely calculating the diffused signal reflected by the hidden object. Methods such as back-projection [9,10], inverse [11], surface normal [12,13], and wave-base [14,15] are used to create an algorithm by analyzing the collected signals. Despite the lack of benchmark datasets, deep-learning models are developing rapidly. Chen et al. [1] collected steady-state data using a conventional camera and a continuous wave laser. Then, the 3D image of the hidden scene is reconstructed using the end-to-end deep-learning network. In addition, static synthetic data has been used to improve the model's performance. Chopite et al. [2] used 3D and 2D encoders to reconstruct the hidden scene. Other end-to-end deep-learning methods included extracting features from multiple layers using deep matrix factorization [16], deep inverse correlation approach (deep inverse), deep matrix factorization, and deep inverse correlation approach. Some studies [17] addressed the noise problem related to correlation using correlography, and [18,19] proposed a feature embedding learning method suitable for hidden scene reconstruction. In addition, there are tasks for identification [20,21] and pose estimation [22] in hidden scenes.

2.2. Acoustic Non-Line-of-Sight Imaging

Optical systems are mainly used for NLOS image reconstruction because they focus on the signal of hidden objects. However, optical equipment that uses light sources, such as lasers, requires specialized knowledge and is expensive. Therefore, Lindell et al. [8] proposed using acoustic equipment to reconstruct the hidden scene.

For acoustic NLOS, a physical model inspired by a geotomography that analyzes the time when the P- and S-waves of the seismic waves reach the observatory is designed. The acoustic equipment emitted linear acoustic chirps (20 Hz–20 kHz) and recorded multi-bounce sounds using speaker and microphone arrays. On the other hand, we simultaneously emitted signals, thus reducing the scanning time. Subsequently, we propose a deep learning model suitable for interfered echoes.

2.3. Echo-Visual Learning

Existing audio-visual learning approaches use sounds passively. Echo-visual Learning, in contrast, uses methods that actively generate sounds. This approach outperformed the conventional audio-visual methods. Echo-visual methods use sounds in fields that require spatial information, such as depth measurements, SLAM, and floor planning. Visualechoes [6] enabled monocular depth estimation, surface normal estimation, and visual navigation in 3D indoor scene environments. Parida et al. [7] estimated depth maps using multi-modal data (RGB images, echoes, and materials of objects) from indoor scenes. Purushwalkam et al. [23] reconstructed the floor plan of the invisible area using echoes. Batvision [5] used both vision and echoes to train, and in the test phase, they estimated depth using echoes only.

3. Approach

3.1. Dataset

We built an experimental environment hidden from the observer’s point of view for data collection. Our data acquisition process and system are as follows. First, we mounted 8 speakers and microphone arrays in the linear stage. Then, The speakers emitted chirp signals simultaneously. The echoes that bounce off a hidden object are recorded using microphones. Subsequently, the acoustic signals are repeatedly acquired while moving horizontally at 10 cm intervals at 8 positions. At the end of the collection process, we obtained 64 echoes and the ground-truth depth map. We repeated the collection process by adjusting the hidden object positions and angles.

The objects included a wooden alphabet, a sign, a plastic mannequin, and an iron fire extinguisher, as shown in Figure 3. In the case of mannequins, the posture is changed, or accessories are attached (e.g., backpacks, baskets). We collected a total of 5376 data from 16 classes. For each class, 336 data are collected from various angles and positions. We configured a speaker (SoundStream LX.402) that could output 70 Hz to 20 kHz and an electret measurement microphone (Dayton Audio EMM-6) that could receive a wide bandwidth and flat without emphasizing a specific band. In addition, an 8-channel audio interface (Behringer UMC1820) and a 12-channel power amplifier (Dayton Audio MA1240a) are used. The linear stage is made to order, as shown in Figure 4. The reflection is dispersed if the reflective surface’s size is comparable to or less than the sound wave’s wavelength. Therefore, rather than employing pure tones, we transmitted linear frequency chirp signals with a duration of 0.1 s, ranging from 20 Hz to 20 kHz. We recorded 0.5 s at a sampling rate of 48 kHz. At a distance of 5 m, we measured the chirp signal volume as approximately 70 dB SPL. The total scan time is approximately 45 s.

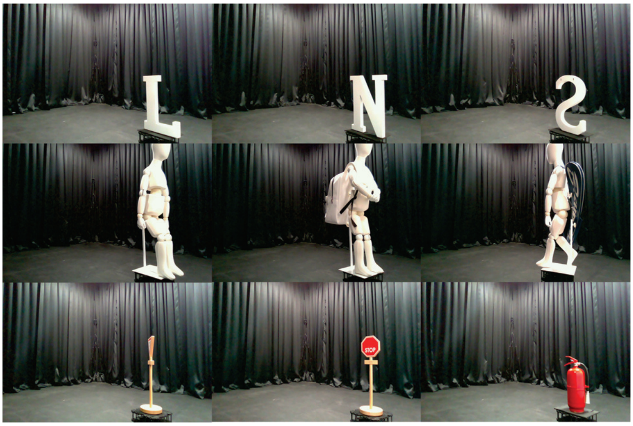


Figure 3. Examples of objects used for data collection.

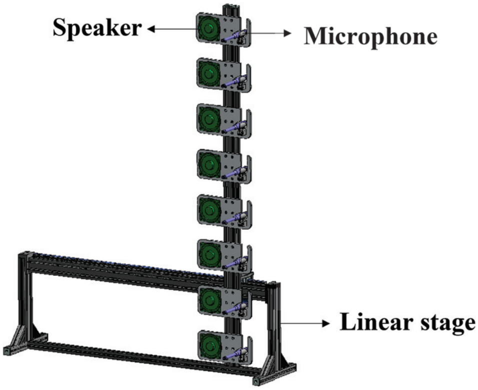


Figure 4. Illustration of our linear stage: a speaker and microphone arrays arranged vertically.

The experiment space is 6.52 m wide and 8.51 m long. The hidden objects are within a 3 m × 3 m area. The minimum and maximum distances between the hidden objects and acoustic equipment are 3 m and 6 m, respectively. We do not establish a soundproof environment. Therefore, we recorded echoes reflected from the walls, floors, ambient noise, and hidden objects. Following data collection, we used Turtle Bot 3 to move the object at various distances and angles.

3.2. Observation Model

Echolocation is achieved by emitting short sounds and perceiving returning echoes. It is similar to the mechanism of active sonar. We employed a frequency-modulated continuous-wave sonar system with MIMO arrays. As a linear frequency sweep, the proposed acoustic sonar transmitted signals $g(t)$ are as follows:

$$g(t) = \sin\left[\phi_0 + 2\pi\left(\frac{c}{2}t^2 + f_0 t\right)\right] \tag{1}$$

where ϕ is the initial phase at time $t = 0$ and c is the chirp rate, which is assumed to be constant:

$$c = \frac{f_1 - f_0}{T} \tag{2}$$

where f_0 is the starting frequency at time $t = 0$ and f_1 is the final frequency T is the time to sweep from f_0 to f_1.

The transmitted signals $g(t)$ passed through the relay wall-hidden object-relay wall to reach the microphones. The received $r(t)$ signals can be expressed in terms of time delay, noise, and transmission loss of the original signals, such that:

$$r(t) = \sum_i \alpha g_i(t - T_i) + n_i(t) \tag{3}$$

where the attenuation transmission loss is denoted by α, the range delay time by T, and the noise is by $n(t)$.

3.3. Reconstruction Model

We aimed to reconstruct a depth map of an object hidden behind walls. We designed a three-part module consisting of an encoder, generator, and discriminator to learn depth map representation from echoes. Figure 5 illustrates the architecture of the proposed model. The proposed model reconstructed the hidden object using echoes. Each echo is collected at different locations. Thus, although they are related, each has independent data. If the features of 64 echoes are extracted simultaneously, extracting the sparse information required for object reconstruction would be difficult. Therefore, we extracted the features necessary for reconstruction using a hierarchical encoder structure. The hierarchical encoder extracted and merged the features by grouping them along the y-axis of the speaker and microphone array. Then, the generator reconstructs the depth map of the hidden object using extracted features, and the quality of the reconstructed depth map is enhanced using a discriminator. Below, we provide details about the modules.

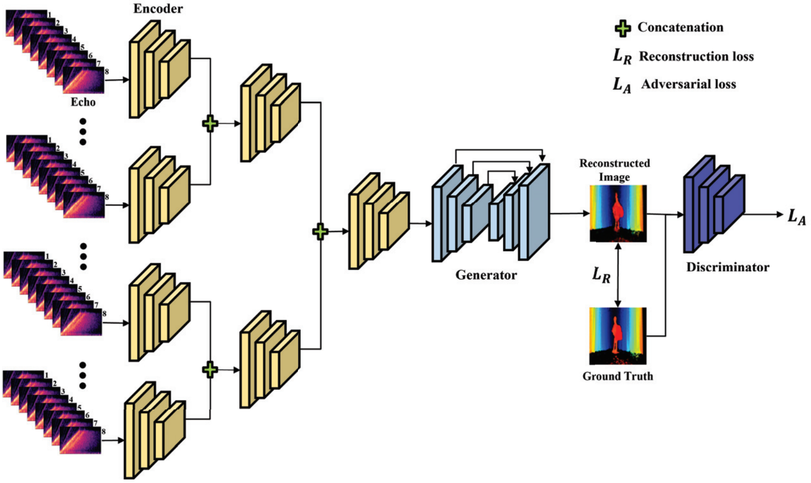


Figure 5. The model generates the depth map from the echoes. The encoder hierarchically extracts hidden object features from the echoes. Then extracted features are concatenated and then reshaped. The generator used the features extracted from the echoes to create a depth map. The discriminator compares the generated depth map with the ground-truth.

3.3.1. Hierarchical Feature Extraction

The encoder learned the depth map representations from echoes. For depth map reconstruction, 64 echoes are input. We divided the 64 echoes into 8 groups along the y-axis. Each group then passed through the first encoder and merged with the adjacent groups to form 4 groups. In this way, features are extracted by hierarchically merging the features in the order of the 8-4-2-1 group.

Each encoder has the same structure, consisting of CNN, convolutional block attention module (CBAM) [24], and squeeze-and-excitation (SE-Net) attention module [25], as shown in Figure 6. The CBAM is a channel and spatial attention module, and the formula is expressed as follows:

$$Channel(X) = Sigmoid(MLP(AvgPool(X) + MLP(MaxPool(X)) \tag{4}$$

Channel attention focused on what is important for a given input. The input feature map X is reshaped into $C \times 1 \times 1$. Then, integrate the spatial information of the feature map using average-pooling and max-pooling, respectively. Each pooling is passed to the multilayer perceptron (MLP) to generate an attention map, following which the two are integrated to create the final channel attention map.

$$Spatial(X) = Sigmoid(Conv(AvgPool(X); MLP(MaxPool(X)) \tag{5}$$

The spatial attention module enabled focusing on the location of significant information. In the feature map generated by multiplying the channel attention map with the input feature map, we concatenated the two values generated by applying max-pooling and average-pooling around the channels. Here, a convolution operation is applied to generate a spatial attention map. The CBAM, which sequentially applied channel and spatial attention, revealed what and where to focus and emphasized the channel with important information in the entire channel while simultaneously focusing on the required area on the feature map.

The SE-Net consisted of a block of squeeze operations, which extracted global information by compressing feature map information, and excitation operation, which adjusted the relative importance of each channel. The squeeze-and-excitation (SE) block is expressed as follows:

$$SE(X) = Sigmoid(MLP(GlobalAveragePool(X))) \tag{6}$$

The features extracted through the last encoder module are converted into one-dimensional vectors through flattening. Then, we reshaped the encoded vectors $H \times W$, where H and W are the height and width of the desired output image, respectively.

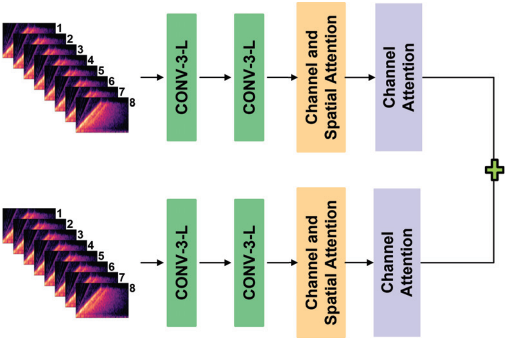


Figure 6. Each encoder consisted of two convolutional layers, CBAM and SE-Net.

3.3.2. Generator

The generator created a depth map from the encoded vectors. We used attention U-Net [26], which can bypass the bottleneck. We input encoded feature maps $H \times W$ into the generator and estimated the depth map $D_i \in \mathbb{R}^{W \times H}$. Attention, which reduces the weight of the background by focusing on the hidden object, is applied to the expanding path of the U-Net (Figure 7). It is defined as follows:

$$q^l_{att} = \psi^T \left(\sigma_1 \left(W^T_x x^l_i + W^T_g g_i + b_g \right) \right) + b_\psi \tag{7}$$

$$\alpha_i^l = \sigma_2\left(q_{att}^l\left(x_i^l, g_i; \Theta_{att}\right)\right) \tag{8}$$

where x^l denotes the output of the contracting path, g denotes the gating signals, b_g, and b_ψ are the bias terms. The terms σ_1 and σ_2 denote ReLU and sigmoid, respectively. This attention is performed immediately before the concatenation procedure.

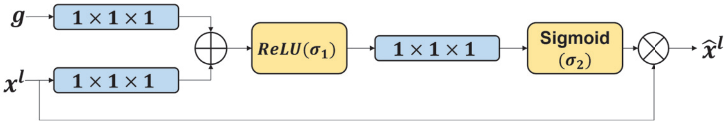


Figure 7. Additive attention gate (AG) of UNet. The separate $1 \times 1 \times 1$ convolutions are used for the input features (x_l) and the gating signal. Following that, the features are added and subjected to several linear transformations, including ReLU activation (σ_1), $1 \times 1 \times 1$ convolution, and sigmoid activation (σ_2).

3.3.3. Discriminator

The discriminator is a network that determines whether an image generated as input is real or fake by comparing it with the ground-truth depth map. The discriminator structure of Pix2Pix [27] is used. The details of the layers are listed in Table 1.

Table 1. Discriminator layer configuration.

Layer	# of Filters	Filter Size	Stride	Padding
Conv1	64	4	2	1
Conv2	128	4	2	1
Conv3	256	4	2	1
Conv4	512	4	2	1
Conv5	1024	4	2	1
Conv6	1	4	2	0

3.3.4. Loss Function

The network is trained using the L1 errors. The loss is specified as follows:

$$\min_G \max_D \frac{1}{2}\mathcal{L}_{GAN}(D) + \mathcal{L}_{GAN}(G) + \lambda\mathcal{L}_{L_1}(G) \tag{9}$$

where λ is the weight factor. We disregard regions that are not defined on the ground-truth depth map.

4. Experiments

4.1. Implementation Details

We used PyTorch to build the proposed model. Initial decays of β_1 and β_2 for the Adam optimizer are 0.5, 0.999, and a learning rate of 0.0001, respectively. The training is run for approximately one day with 100 epochs on a single NVIDIA V100. Input to the model is converted into a spectrogram of 0.3 s. The input depth map images are resized to 128 × 128. We used a window size of 64 and a fast Fourier transform the size of 512 for all the experiments. In the 16 classes we collected, we used 11 classes for training and the remaining 5 classes for testing unseen situations. We divided the training and validation data into a ratio of 8:2.

4.2. Evaluation Metrics

We used evaluation methods frequently used for depth estimation and evaluation methods for measuring the similarity of depth maps. The following p is a pixel, d_p is a ground-truth depth map, $\hat{d_p}$ is a predicted depth map, and T is the total number of pixels with both valid ground-truth and depth map outputs.

- Structural Similarity Index Measure (SSIM) [28]:

$$SSIM(d_p, \hat{d_p}) = I(d_p, \hat{d_p}) \times c(d_p, \hat{d_p}) \times s(d_p, \hat{d_p}) \tag{10}$$

where i, c, and s are luminance, contrast, and structure, respectively.

- Root Mean Square Error (RMSE) [29]:

$$\sqrt{\frac{1}{T}\sum_p (d_p - \hat{d_p})^2} \tag{11}$$

- Absolute Relative Error (Abs Rel) [30]:

$$\frac{1}{T}\sum_p \frac{\left|d_p - \hat{d_p}\right|}{d_p} \tag{12}$$

- Accuracy under threshold [31]:

$$max\left(\frac{\hat{d_p}}{d_p}, \frac{d_p}{\hat{d_p}}\right) = \delta < threshold \tag{13}$$

where the threshold is 1.25, 1.25^2, and 1.25^3, respectively.

4.3. Results

In acoustic NLOS, there is no existing method that uses deep learning. Furthermore, it is difficult to compare the existing proposed physical model with any extant approach because of our deployed unique data collection process. Thus, we compared the encoders of the three structures, as shown in Figure 8 below. We compared three types of encoders: plain, split, and hierarchical encoders. A plain encoder is a general structure that extracts features without separating the 64 channels of echoes. The split encoder divided the 64 channels into eight groups to extract and merge the features. The hierarchical encoder is a structure that extracts features by dividing them into groups and hierarchically merging them.

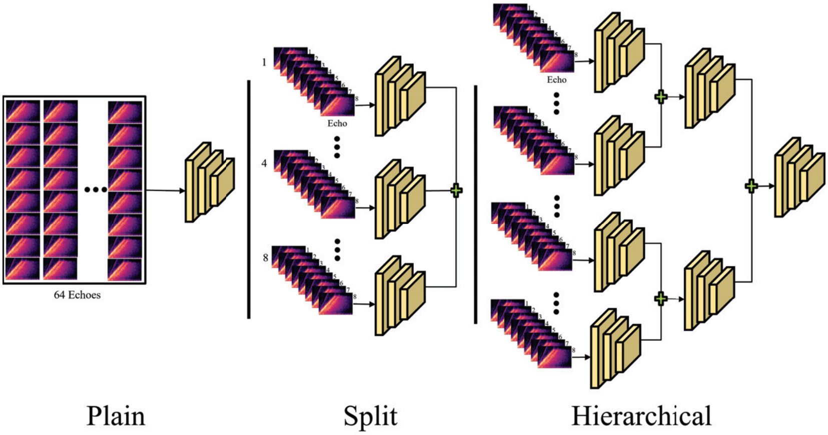


Figure 8. Encoder architecture comparison.

4.3.1. Quantitative Results

Table 2 compares the three types of encoders. The models used in the plain encoder structure included CNN, 3D CNN [32], ResNet [33], SE-ResNet [25], ResNext [34], and Batvision [5]. The CNN model consisted of six convolutional layers and two fully connected layers. The 3D CNN model consisted of eight layers and two fully connected layers, with the y-axis where is the channel and the x-axis is the depth. ResNet, SE-ResNet, and ResNext used 50 layers. Batvision is a model reconstruction using only echoes in a line-of-sight situation. In the split encoder, echoes are grouped based on the y-axis, and CNN, 3D CNN, and ResNet are used as models, and the structure is the same as that of the plain encoder. In the hierarchical encoder, we compared the CNN with our proposed model. The structure of the generator and discriminator is fixed.

Table 2. Quantitative results comparison.

Encoder	Model	SSIM (↓)	RMSE (↓)	Abs Rel (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
Plain	CNN	0.219	0.235	2.094	0.460	0.603	0.692
	3DCNN	0.205	0.235	2.031	0.486	0.621	0.706
	ResNet	0.222	0.257	2.324	0.452	0.590	0.677
	SE-ResNet	0.230	0.249	2.282	0.456	0.595	0.682
	ResNext	0.228	0.249	2.348	0.466	0.602	0.688
	Batvision	0.216	0.236	2.519	0.453	0.594	0.681
	Ours	0.224	0.251	2.187	0.457	0.595	0.682
Split	CNN	0.213	0.250	2.171	0.460	0.597	0.684
	3DCNN	0.231	0.254	2.428	0.462	0.599	0.684
	ResNet	0.221	0.256	2.402	0.473	0.612	0.698
	Ours	0.217	0.263	2.690	0.462	0.595	0.678
Hierarchical	CNN	0.221	0.249	2.396	0.463	0.599	0.685
	Ours	0.188	0.230	1.908	0.490	0.627	0.713

In the plain encoder, the 3D CNN model exhibited the best performance. These results demonstrate that the 3D CNN, which can consider the spatial relationship, has superior reconstruction performance compared to the other models. Then, the simple CNN model performed better on the data than the other models, except for the 3D CNN. These results show that the sparse information required for reconstruction disappeared as the layer deepened. Generally, the performance of a model grows with increasing depth. However, it can be observed that the performance of the other models degraded because the plain encoder does not consider the data characteristics.

The performance of the 3D CNN model is worse with the split encoder than with the plain encoder. The 3D CNN model can consider spatial relations without grouping the echoes, but the split encoder structure does not consider the relationship between each group by grouping the echoes. Thus, in this structure, CNN exhibited the best performance. The split encoder used each group independently to extract and merge features. Therefore, it is difficult to consider the spatial relationships of the other groups.

Our proposed hierarchical encoder structure and model outperformed all the other models based on all the evaluation matrices. We showed that our attention model is adequate compared to the CNN model in the hierarchy structure. Therefore, the proposed method proved that the hierarchical structure and the proposed model, with its 64 independent echoes, are suitable for feature extraction.

4.3.2. Qualitative Results

Figure 9 compares the depth maps generated by the model in each encoder architecture. The 3D CNN model reconstruction of a plain encoder has an approximate shape but is less detailed than the proposed method. For example, the lower part of L is not reconstructed in the first row, and the mannequin in the third row has no arms. The image yielded by the split encoder CNN model is only partially reconstructed. Our proposed method generated more distinguishable shapes than the other encoder architecture models.

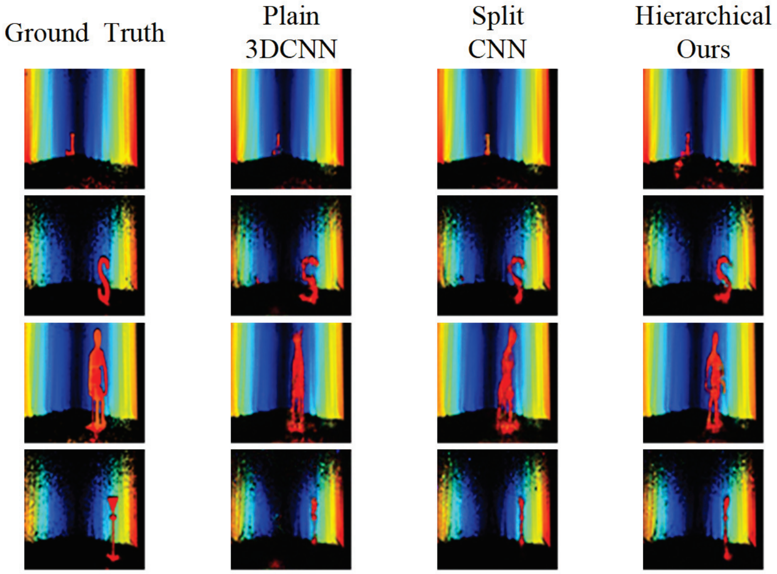


Figure 9. Comparison of image reconstruction according to encoder architecture.

In the test, as shown in Figure 10, the wooden L and N are reconstructed approximately such that they are recognizable. However, owing to its relatively complex shape, S is not recognizable. Mannequins are recognizable in all classes. However, if the details, arms, and legs are slightly different, they are not accurately reconstructed. The symbol has the shape of a square, hexagon, or circle. Signs at a close distance indicated the shape better than they do at a distance. With increasing distance, the object's reconstruction is less precise. However, although mannequins' small body parts (e.g., ears, finger) is difficult to reconstruct, body parts with large surface areas are well reconstructed at a distance. Furthermore, letters or symbols with a small surface area do not recover well as the distance increases.

As shown in Figure 11, unseen data are not used in training. The data used in training are similar but not the same (e.g., the letters N, L, and S are used for training, but O is only used for unseen). When comparing unseen data that are not used in training, the approximate shape of the object is reconstructed, but the details cannot be reconstructed. The large-surface-area mannequins are reconstructed to be recognizable, but the small alphabets and signs are not.

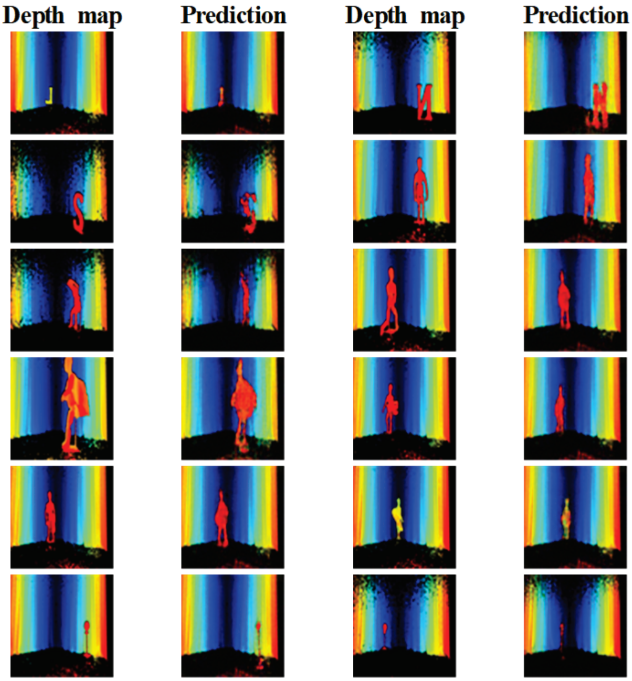


Figure 10. Qualitative result of proposed method.

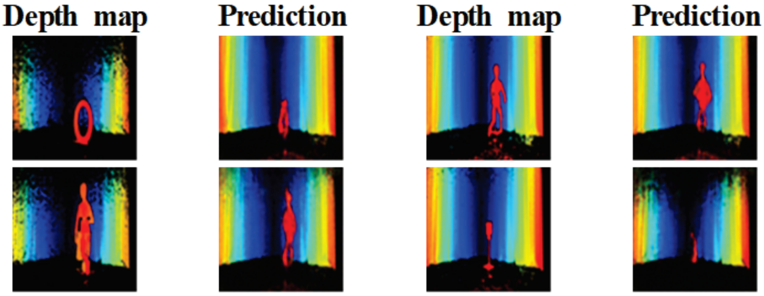


Figure 11. Depth prediction using data not included in the training dataset.

5. Conclusions

In this study, we designed an active sonar system inspired by echolocation, certain animals' innate ability to reconstruct an object's image in a hidden scene. The acoustic system emitted sound waves and used reflections to identify hidden object positions and approximate shapes. Our system is designed using the MIMO method, which reduced the required data collection time from 4.5 m, as required by the conventional method, to 40 s. Furthermore, we do not use the sound-absorbing material used in the previous method. It is challenging to reconstruct hidden objects from mutually interfering echoes using conventional physical methods; therefore, we proposed a deep-learning approach. Specifically, we created a hierarchical encoder and model to extract echoes effectively with 64 independent channels. Consequently, our model outperformed models of encoders with other structures.

Author Contributions: Conceptualization, S.J. and K.K.; methodology, S.J.; software, S.J.; validation, S.J.; formal analysis, S.J.; investigation, S.J.; resources, S.J.; data curation, S.J. and U.-H.S.;

writing—original draft preparation, S.J.; writing—review and editing, S.J. and K.K.; visualization, S.J.; supervision, K.K.; project administration, K.K.; funding acquisition, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2020-0-00973).

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.

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

Abbreviations

The following abbreviations are used in this manuscript:

NLOS	Non-Line-of-Sight
SPAD	Single-Photon Avalanche Diode
ToF	Time-of-Flight
SIMO	Single Input, Multiple Outputs
MIMO	Multiple Inputs, Multiple Outputs
FMCW	Frequency-Modulated Continuous-Wave
SNR	Signal-to-Noise Ratio
FFT	Fast Fourier Transform
CNN	Convolutional Neural Networks

References

1. Chen, W.; Daneau, S.; Mannan, F.; Heide, F. Steady-state non-line-of-sight imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
2. Chopite, J.G.; Hullin, M.B.; Wand, M.; Iseringhausen, J. Deep non-line-of-sight reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020.
3. Tancik, M.; Satat, G.; Raskar, R. Flash photography for data-driven hidden scene recovery. *arXiv* **2018**, arXiv:1810.11710.
4. Rosenblum, L.D.; Gordon, M.S.; Jarquin, J. Echolocating distance by moving and stationary listeners. *Ecol. Psychol.* **2000**, *12*, 181–206. [CrossRef]
5. Christensen, J.H.; Hornauer, S.; Yu, S.X. BatVision: Learning to See 3D Spatial Layout with Two Ears. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–15 June 2020.
6. Gao, R.; Chen, C.; Al-Halah, Z.; Schissler, C.; Grauman, K. Visualechoes: Spatial image representation learning through echolocation. In Proceedings of the European Conference on Computer Vision (ECCV), Virtual, 23–28 August 2020.
7. Parida, K.K.; Srivastava, S.; Sharma G. Beyond image to depth: Improving depth prediction using echoes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
8. Lindell, D.B.; Wetzstein, G.; Koltun, V. Acoustic non-line-of-sight-imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
9. Velten, A.; Willwacher, T.; Gupta, O.; Veeraraghavan, A.; Bawendi M.G.; Raskar, R. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nat. Commun.* **2012**, *3*, 745. [CrossRef] [PubMed]
10. Arellano, V.; Gutierrez, D.; Jarabo, A. Fast back-projection for non-line-of sight reconstruction. *Opt. Express* **2017**, *25*, 11574–11583. [CrossRef]
11. O'Toole, M.; Lindell, D.B.; Wetzstein, G. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature* **2018**, *555*, 338–341. [CrossRef] [PubMed]
12. Heide, F.; O'Toole, M.; Zang, K.; Lindell, D.B.; Diamond, S.; Wetzstein, G. Non-line-of-sight imaging with partial occluders and surface normals. *ACM Trans. Graph.* **2019**, *38*, 1–10. [CrossRef]
13. Xin, S.; Sankaranarayanan, A.C.; Narasimhan, S.G.; Gkioulekas, I. A theory of fermat paths for non-line-of-sight shape reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
14. Lindell, D.B.; Wetzstein, G.; O'Toole, M. Wave-based non-line-of-sight imaging using fast fk migration. *ACM Trans. Graph.* **2019**, *38*, 1–13. [CrossRef]
15. Liu, X.; Bauer, S.; Velten, A. Phasor field diffraction based reconstruction for fast non-line-of-sight imaging systems. *Nat. Commun.* **2020**, *11*, 1–13. [CrossRef] [PubMed]

16. Aittala, M.; Sharma, P.; Murmann, L.; Yedidia, A.B.; Wornell, G.W.; Freeman, W.T.; Durand, F. Computational mirrors: Blind inverse light transport by deep matrix factorization. *arXiv* **2019**, arXiv:2005.00007.
17. Metzler, C.A.; Heide, F.; Rangarajan, P.; Balaji, M.M.; Viswanath, A.; Veeraraghavan, A.; Baraniuk, R.G. Deep-inverse correlography: Towards real-time high-resolution non-line-of-sight imaging. *Optica* **2020**, *7*, 63–71. [CrossRef]
18. Chen, W.; Wei, F.; Kutulakos, K.N.; Rusinkiewicz, S.; Heide, F. Learned feature embeddigns for non-line-of-sight imaging and recognition. *ACM Trans. Graph.* **2020**, *39*, 1–18.
19. Zhu, D.; Cai, W. Fast non-line-of-sight imaging with two-step deep remapping. *ACS Photonics* **2021**, *9*, 2046–2055. [CrossRef]
20. Caramazza, P.; Boccolini, A.; Buschek, D.; Hullin, M.; FHigham, C.; Henderson, R.; Murray-Smith, R.; Faccio, D. Neural network identification of people hidden from view with a single-pixel, single-photon detector. *Sci. Rep.* **2018**, *8*, 11945. [CrossRef] [PubMed]
21. Musarra, G.; Caramazza, P.; Turpin, A.; Lyons, A.; FHigham, C.; Murray-Smith, R.; Faccio, D. Detection, identification and tracking of objects hidden from view with neural networks. *Adv. Photon Count. Tech. XIII* **2019**, *10978*, 109803.
22. Isogawa, M.; Yuan, Y.; O'Toole, M.; Kitani, K.M. Optical non-line-of-sight physics-based 3d human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virture, 14–19 June 2020.
23. Purushwalkam, S.; Gari, S.V.A.; Ithapu, V.K.; Robinson, C.S.P.; Gupta, A.; Grauman, K. Audio-visual floorplan reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021.
24. 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.
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
26. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B. Attention u-net: Learning where to look for the pancreas. In Proceedings of the International Conference on Medical Imaging with Deep Learning (MIDL), Amsterdam, The Netherlands, 4–6 July 2018.
27. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
28. 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] [PubMed]
29. Li, C.; Kowdle, A.; Saxena, A.; Chen, T. Towards holistic scene understanding: Feedback enabled cascaded classification models. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 6–11 December 2010.
30. Saxena, A.; Sun, M.; Ng, A.Y. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 824–840. [CrossRef] [PubMed]
31. Ladicky, L.; Shi, J.; Pollefeys M. Pulling things out of perspective. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
32. 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 (ICCV), Santiago, Chile, 7–13 December 2015.
33. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
34. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

sensors

Article

Fully Cross-Attention Transformer for Guided Depth Super-Resolution

Ido Ariav * and Israel Cohen *

Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion—Israel Institute of Technology, Haifa 3200003, Israel
* Correspondence: idoariav@campus.technion.ac.il (I.A.); icohen@ee.technion.ac.il (I.C.)

Abstract: Modern depth sensors are often characterized by low spatial resolution, which hinders their use in real-world applications. However, the depth map in many scenarios is accompanied by a corresponding high-resolution color image. In light of this, learning-based methods have been extensively used for guided super-resolution of depth maps. A guided super-resolution scheme uses a corresponding high-resolution color image to infer high-resolution depth maps from low-resolution ones. Unfortunately, these methods still have texture copying problems due to improper guidance from color images. Specifically, in most existing methods, guidance from the color image is achieved by a naive concatenation of color and depth features. In this paper, we propose a fully transformer-based network for depth map super-resolution. A cascaded transformer module extracts deep features from a low-resolution depth. It incorporates a novel cross-attention mechanism to seamlessly and continuously guide the color image into the depth upsampling process. Using a window partitioning scheme, linear complexity in image resolution can be achieved, so it can be applied to high-resolution images. The proposed method of guided depth super-resolution outperforms other state-of-the-art methods through extensive experiments.

Keywords: super-resolution; deep learning; depth maps; attention; multimodal; transformers

Citation: Ariav, V.; Cohen, I. Fully Cross-Attention Transformer for Guided Depth Super-Resolution. *Sensors* **2023**, *23*, 2723. https://doi.org/10.3390/s23052723

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 16 February 2023
Revised: 26 February 2023
Accepted: 27 February 2023
Published: 2 March 2023

1. Introduction

High-resolution (HR) depth information of a scene plays a significant part in many applications, such as 3D reconstruction [1], driving assistance [2], and mobile robots. Nowadays, depth sensors such as LIDAR or time-of-flight cameras are becoming more widely used. However, they often suffer from low spatial resolution, which does not always suffice for real-world applications. Thus, ongoing research has been done on reconstructing a high-resolution depth map from a corresponding low-resolution (LR) counterpart in a process termed depth super-resolution (DSR).

The LR depth map does not contain the fine details of the HR depth map, so reconstructing the HR depth map can be challenging. Bicubic interpolation, for example, often produces blurry depth maps when upsampling the LR depth, which limits the ability to, e.g., separate between different objects in the scene.

In recent years, many learning-based approaches based on elaborate convolutional neural network (CNN) architectures for DSR were proposed [3–7]. These methods surpassed the more classic approaches such as filter-based methods [8,9], and energy minimization-based methods [10–12] in terms of computation speed and the quality of the reconstructed HR information. Although CNN-based methods improved the performance significantly compared with traditional methods, they still suffer from several drawbacks. To begin with, feature maps derived from a convolution layer have a limited receptive field, making long-range dependency modeling difficult. Second, a kernel in a convolution layer operates similarly on all parts of the input, making it content-independent and likely not the optimal choice. In contrast to CNN, transformers [13] have recently shown promising results in several vision-related tasks due to their use of attention. The attention mechanism enables the transformer to operate in a content-dependent manner, where each input part is treated differently according to the task.

LR depth information is often accompanied by HR color or intensity images in real-life situations. Thus, numerous methods proposed to use this HR image to guide the DSR process [3,4,7,14–23] since the HR image might provide some additional information that does not exist in the LR depth image, e.g., the edges of a color image can be used to identify discontinuities in a reconstructed HR depth image. However, one major limitation, termed texture-copying, still exists in these guided DSR methods. Texture copying may occur when intensity edges do not correspond to depth discontinuities in-depth maps, for example, a flat and highly textured surface. Consequently, the reconstruction of HR depth is then degraded due to the over-transfer of texture information.

This paper proposes a novel, fully transformer-based architecture for guided DSR. Specifically, the proposed architecture consists of three modules: shallow feature extraction, deep feature extraction and fusion, and an upsampling module. In this paper, we term the feature extraction and fusion module the cross-attention guidance module (CAGM). The shallow feature extraction module uses convolutional layers to extract shallow features from LR depth and HR color images, which are directly fed to the CAGM to preserve low-frequency information. Next, several transformer blocks are stacked to form the CAGM, each operating in non-overlapping windows from the previous block. Guidance from the color image is introduced via a cross-attention mechanism. In this manner, guidance from the HR color image is seamlessly integrated into the deep feature extraction process. This enables the network to focus on salient and meaningful features and enhance the edge structures in the depth features while suppressing textures in the color features. Moreover, contrary to CNN-based methods, which can only use local information, transformer blocks can exploit the input image's local and global information. This allows learning of structure and content from a wide receptive field, which is beneficial for SR tasks [24]. As a final step, shallow and deep features are fused in the upsampling module to reconstruct HR depth. Section 4 shows that the proposed architecture provides better visual results with sharper boundaries and better root mean square error (RMSE) values than competing guided DSR approaches. We also show how the proposed architecture helps to mitigate the texture-copying problem of guided DSR. The proposed architecture is shown in Figure 1.

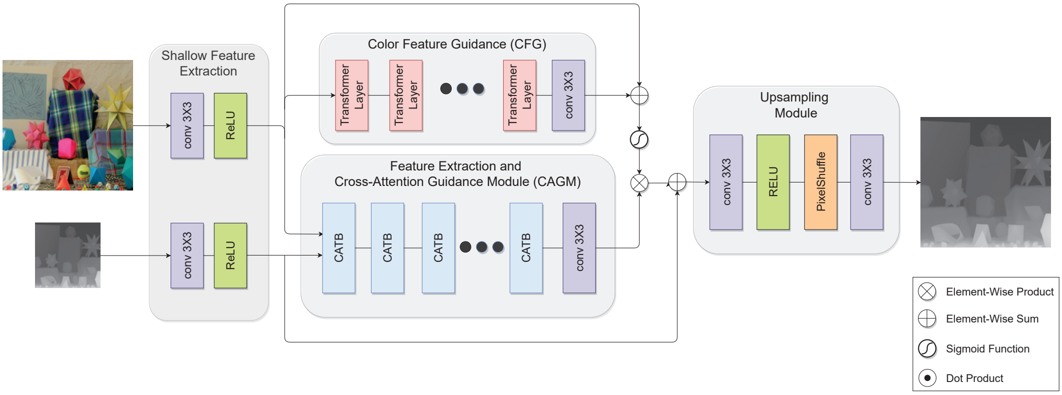


Figure 1. The proposed FCTN architecture for guided depth SR with a 2× upsampling factor.

Our main contributions are as follows: (1) We introduce a transformer-based architecture with a novel guidance mechanism that leverages cross-attention to seamlessly integrate guidance features from a color image to the DSR process. (2) Linear memory constraints make the proposed architecture applicable even for large inputs. (3) This architecture is not limited to a fixed input size, so it can be applied to a variety of real-world problems. (4) Our system achieves state-of-the-art results on several depth-upsampling benchmarks.

The remainder of this paper is organized as follows. A summary of related work is presented in Section 2. We describe our architecture for guided DSR in Section 3. Section 4 reports the results of extensive experiments conducted on several popular DSR datasets. Additionally, an ablation study is conducted. We conclude and discuss future research directions in Section 5.

2. Related Work

2.1. Guided Depth Map Super-Resolution

A number of methods for reconstructing the HR depth map only from LR depth have been proposed in earlier works for single depth map SR. ATGV-Net was proposed by [5] combining a deep CNN in tandem with a variational method designed to facilitate the recovery of the HR depth map. Reference [25] modeled the mapping between HR and LR depth maps by utilizing densely connected layers coupled with a residual learning model. Auxiliary losses were tabulated at various scales to improve training.

However, it is pertinent to note that in most real-life scenarios, the LR depth image is coupled with a HR intensity image. Recently, several methods have been proposed to improve depth image quality, relying on the HR intensity image to guide the upsampling process. We group these methods under four sub-categories: filtering-based methods [26–28], global optimization-based methods [10–12,16,29–34], sparse representation-based methods [14,15], and deep learning-based methods [3,4,7,17–23,35–40], which are the focus of this paper.

Notable amongst the more classical works are [10], which formulated the upsampling of depth as a convex optimization problem. The upsampling process was guided by a HR intensity image. A bimodal co-sparse analysis was presented in [14] to describe the interdependency of the registered intensity and depth information. Reference [15] proposed a multi-scale dictionary as a method for depth map refinement, where local patches were represented in both depth and color via a combination of select basis functions.

Deep learning methods for SR of depth images have gained increasing attention due to recent success in SR of color images. A fully convolutional network was proposed in [35] to estimate the HR depth. To optimize the final result, this HR estimation was fed into a non-local variational model. Reference [4] proposed an "MSG-Net", in which both LR (depth) and HR (color) features are combined within the high-frequency domain using a multi-scale fusion strategy. Reference [3] proposed extracting hierarchical features from depth and color images by building a multi-scale input pyramid. The hierarchical features are further concatenated to facilitate feature fusion, whilst the residual map between the reconstructed and ground truth HR depth is learned with a U-Net architecture. Reference [37] proposed another multi-scale network in which the LR depth map upsampling, guided by the HR color image, was performed in stages. Global and local residual learning is applied within each scale. Reference [17] proposed a cosine transform network in which features from both depth and color images were extracted using a semi-coupled feature extraction module. To improve depth upsampling, edges were highlighted by an edge attention mechanism operating on color features. Reference [19] proposed to use deformable convolutions [41] for the upsampling of depth maps, using the features of the HR guidance image to determine the spatial offsets. Reference [42] also applied deformable convolutions to enhance depth features by learning the corresponding feature of the high-resolution color image. An adaptive feature fusion module was used to fuse different level features adaptively. A network based on residual channel attention blocks was proposed in [20], where feature fusion blocks based on spatial attention were utilized to suppress texture-copying. Reference [21] proposed a progressive multi-branch aggregation design that gradually restores the degraded depth map. Reference [22] proposed separate branches for HR color image and LR depth map. A dual-skip connection structure, together with a multi-scale fusion strategy, allowed for more effective features to be learned. Reference [39] used a transformer module to learn the useful content and structure information from the depth maps and the corresponding color images, respectively. Then, a multi-scale fusion strategy was used to improve the efficiency of color-depth fusion. Reference [43] proposed explicitly incorporating the depth gradient features in the DSR process. Reference [44] proposed PDR-Net, which incorporates an adaptive feature recombination module to adaptively recombine features from a HR color guidance image with features from the LR depth. Then, a multi-scale feature fusion module is used to fuse the recombined features using multi-scale feature distillation and joint attention mechanism. Finally, Reference [23] presented an upsampling method that incorporates the intensity image's high-resolution structural information into a multi-scale residual deep network via a cascaded transformer module.

However, the methods above mostly fuse the guidance features with the depth features using mere concatenation. Moreover, most of these methods rely on CNN for feature extraction, which operates on a limited receptive field and lacks the expressive power of transformers. At the same time, we propose using a CAGM module, which leverages transformers to fuse and extract meaningful features from HR color and LR depth images, resulting in superior results, as shown in Section 4.

2.2. Vision Transformers

Transformers [13] have gained great success across multiple domains recently. Contributing to this success was their inherent attention mechanism, which enables them to learn the long-range dependencies in the data. This success led many researchers to adopt transformers to computer vision tasks, where they have recently demonstrated promising results, specifically in image classification [45–47], segmentation [47,48], and object detection [49,50].

To allow transformers to handle 2D images, an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ is first divided into non-overlapping patches of size (P, P). Each patch is flattened and projected to a d-dimensional vector via a trainable linear projection, forming the patch embeddings $\mathbf{X} \in \mathbb{R}^{N \times d}$ where H, W are the height and width of the image, respectively, C is the number of channels, and $N = H \times W / P^2$ is the total number of patches. Finally, N is the effective input sequence length for the transformer encoder. Patch embeddings are enhanced with position embeddings to retain 2D image positional information.

In [13], a vanilla vision transformer encoder is constructed by stacking blocks of multi-head self-attention (MSA) and MLP layers. A residual connection is applied after every block, and layer normalization (LN) before every block. Given a sequence of embeddings $\mathbf{X} \in \mathbb{R}^{N \times d}$ with dimension d as input, a MSA block produces an output sequence $\bar{\mathbf{X}} \in \mathbb{R}^{N \times d}$ via

$$\begin{aligned} &\mathbf{Q} = \mathbf{X}\mathbf{W}_Q \,, \mathbf{K} = \mathbf{X}\mathbf{W}_K \,, \mathbf{V} = \mathbf{X}\mathbf{W}_V \\ &\mathbf{A} = \mathrm{Softmax}(\mathbf{Q}\mathbf{K}^T / \sqrt{\mathbf{d}}) \\ &\bar{\mathbf{X}} = \mathbf{A}\mathbf{V} \end{aligned} \tag{1}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable matrices of size $d \times d$ that project the sequence $\mathbf{X}$ into keys, queries, and values, respectively. $\bar{\mathbf{X}}$ is a linear combination of all the values in $\mathbf{V}$ weighted by the attention matrix $\mathbf{A}$. In turn, $\mathbf{A}$ is calculated from similarities between the keys and query vectors.

Transformers derive their modeling capabilities from computing self-attention $\mathbf{A}$ and $\bar{\mathbf{X}}$. Since self-attention has a quadratic cost in time and space, it cannot be applied directly to images as N quickly becomes unmanageable. As a result of this inherent limitation, modality-aware sequence length restrictions have been applied to preserve the model's performance while restricting sequence length. Reference [45] showed that a transformer architecture could be directly applied to medium-sized image patches for different vision tasks. The aforementioned memory constraints are mitigated by this local self-attention.

Although the above self-attention module can effectively exploit intra-modality relationships in the input image, in a multi-modality setting, the inter-modality relationships, e.g., the relationships between different modalities, also need to be explored. Thus, a cross-attention mechanism was introduced in which attention masks from one modality highlight the extracted features in another. Contrary to self-similarity, wherein query, key, and value are based on similarities within the same feature array, in cross-attention, keys, and values are calculated from features extracted from one modality, while queries are calculated from the other. Formally, a MSA block using cross-attention is given by

$$\mathbf{Q} = \hat{\mathbf{X}}\mathbf{W}_Q\,, \mathbf{K} = \mathbf{X}\mathbf{W}_K\,, \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{2}$$

where $\mathbf{X}$ is the input sequence of one modality and $\hat{\mathbf{X}}$ is the input sequence of the second modality. The calculation of attention matrix $\mathbf{A}$ and output sequence $\bar{\mathbf{X}}$ remains the same.

3. Proposed Method

3.1. Formulation

A guided DSR method aims to establish the nonlinear relation between corresponding LR and HR depth maps. The process of establishing this nonlinear relation is guided by a HR color image. We denote the LR depth map as $\mathbf{D}_{\text{LR}} \in \mathbb{R}^{H/s \times W/s}$ and the HR guidance color image as $\mathbf{I}_{\text{HR}} \in \mathbb{R}^{H \times W}$, where s is a given scaling factor. The corresponding HR depth map $\mathbf{D}_{\text{HR}} \in \mathbb{R}^{H \times W}$ can be found from:

$$\mathbf{D}_{\text{HR}} = \hat{\mathbf{F}}(\mathbf{D}_{\text{LR}}, \mathbf{I}_{\text{HR}}; \theta) \tag{3}$$

where $\hat{\mathbf{F}}$ represents mapping learned by the proposed architecture, and θ represents the parameters of the learned network. Although the scaling factor s is usually an exponent of 2, e.g., $s = 2, 4, 8, 16$, our upsampling module can perform upsampling for other scaling factors as well, making this architecture flexible enough for real applications.

3.2. Overall Network Architecture

Throughout the remainder of this paper, we denote the proposed architecture as the fully cross-attention transformer network (FCTN). As shown in Figure 1, the proposed architecture consists of three parts: a shallow feature extraction module, a deep feature extraction and guidance module called the cross-attention guidance module (CAGM), and an upsampling module. The CAGM extracts features from the LR depth image and guides the HR intensity image simultaneously.

Before we elaborate on the structure of each module, some significant challenges in leveraging transformers' performance for visual tasks, specifically SR, need to be addressed. First, in real-life scenarios, images can vary considerably in scale. Transformer-based models, however, work only with tokens of a fixed size. Furthermore, to maintain HR information, SR methods avoid downscaling the input as much as possible. Processing HR inputs of this magnitude would be unfeasible for vanilla transformers due to computational complexity as described in Section 2.2.

3.2.1. Shallow Feature Extraction Module

The proposed shallow feature extraction module extracts essential features to be fed to the CAGM. Shallow features are extracted from LR depth and HR color images via a single convolution layer with a kernel size of 3 × 3, followed by an activation function of a rectified linear unit (ReLU). In the experiments, we did not notice any noticeable improvement by using more than a single layer for shallow feature extraction. For shallow feature extraction, incorporating a convolution layer leads to more stable optimization and better results [51–53]. Moreover, the input space can also be mapped to a higher-dimensional feature space d easily.

Specifically, the shallow feature extraction module can be formulated as

$$\mathbf{I}_0 = \sigma(\mathbf{Conv}_3(\mathbf{I}_{\text{HR}})) \tag{4}$$

$$\mathbf{D}_0 = \sigma(\mathbf{Conv}_3(\mathbf{D}_{\text{LR}})) \tag{5}$$

where σ is a ReLU activation function and $\mathbf{Conv}_3(\cdot)$ is a 3×3 kernel.

3.2.2. Deep Feature Extraction and Guidance Module

While shallow features primarily contain low frequencies, deep features recover lost high frequencies. We propose a stacked transformer module that extracts deep features from the LR depth image based on the work of [47]. Self(cross)-attention is computed locally within non-overlapping windows, with complexity linear with image size. Working with large and variable-sized inputs is made feasible due to the aforementioned linear complexity. In addition, we shift the windows partitioning into consecutive layers. Overlapping of the shifted and preceding layer windows causes neighboring patches to gradually merge, and thus modeling power is significantly enhanced. Overall, the transformer-based module can efficiently extract and encode distant dependencies needed for dense tasks such as SR.

In addition, motivated by [54], we employ global and local skip connections. By using long skip connections, low-frequency information can be transmitted directly to the upsampling module, helping the CAGM focus on high-frequency information and stabilize training [51]. Furthermore, it allows the aggregation of features from different blocks by using such identity-based connections.

Besides deep feature extraction, a practical guidance module is also required to enhance the deep features extracted from LR depth and exploit the inter-modality information from the available HR color image. Traditionally, CNN-based methods extract features from the color image and concatenate them with features extracted from the depth image in a separate branch to obtain guidance from the color image. All features handled via this guidance scheme are treated equally in both the spatial and channel domains. Furthermore, CNN-derived feature maps have a limited receptive field, affecting guidance quality. In comparison, we propose providing guidance from the HR color image by incorporating a cross-attention mechanism to the aforementioned feature extraction transformer module. Cross-attention is a novel and intuitive fusion method in which attention masks from one modality highlight the extracted features in another. In this manner, both the inter-modality and intra-modality relationships are learned and optimized in a unified model. Thus, in the proposed CAGM, the feature extraction process from the LR depth and guidance from the HR image are seamlessly integrated into a single branch. Guidance from the HR image is injected into every block in the feature extraction module, providing multi-stage guidance. In particular, guidance provided to the lower-level features passed through the long skip connections ensures that high-resolution information is preserved and passed to the upsampling module. Lastly, by incorporating the guidance in the form of cross-attention, long-range dependencies between the LR depth patches and the guidance image patches can be exploited for better feature extraction.

To exploit the HR information further, we feed the HR intensity image to a second cascaded transformer module termed color feature guidance (CFG) to extract even more valuable HR information. The CFG is based on self-attention only and aims to encode distant dependencies in the HR image. These features are then used to scale the features extracted from the CAGM by element-wise multiplication before feeding them to the upsampling module.

We note that contrary to common practice in vision tasks, no downsampling of the input is done throughout the network. This way, our architecture preserves as much high-resolution information as possible, albeit at a higher computational cost.

Formally, given $\mathbf{I}_0$ and $\mathbf{D}_0$, provided by the shallow feature extraction module as input, the CAGM applies K cross-attention transformer blocks (CATB). Every CATB is constructed from L cross-attention transformer layers (CATL), and a convolutional layer and residual skip connection are inserted at the end of every such block. Finally, a 3×3 convolutional layer is applied to the output of the last CATB. This last convolutional layer improves the later aggregation of shallow and deep features by bringing the inductive bias of the convolution operation into the transformer-based network. Furthermore, the translational equivariance of the network is enhanced. In addition, $\mathbf{I}_0$ is fed to the CFG comprised of $\hat{L}$

transformer layers with self-attention. The CFG output is scaled to $[0,1]$ using a sigmoid function and then used to scale the CAGM output before the upsampling module

The CFG module is formulated as

$$\hat{\mathbf{I}}_{\mathrm{i}} = \mathbf{TL}_i(\hat{\mathbf{I}}_{\mathrm{i}-1}), \quad i = 1\ldots\hat{L} \tag{6}$$

$$\mathbf{F}_{\mathrm{CFG}} = \mathbf{Conv}_3(\hat{\mathbf{I}}_{\hat{L}}) + \mathbf{I}_0 \tag{7}$$

where $\hat{\mathbf{I}}_0 = \mathbf{I}_0$ and TL stands for a vanilla transformer layer with self-attention. Finally, the entire CAGM can be formulated as

$$(\mathbf{I}_{\mathrm{i}}, \mathbf{D}_{\mathrm{i}}) = \mathbf{CATB}_i(\mathbf{I}_{\mathrm{i}-1}, \mathbf{D}_{\mathrm{i}-1}), \quad i = 1\ldots K \tag{8}$$

$$\mathbf{F}_{\mathrm{CAGM}} = \mathbf{Conv}_3(\mathbf{CATB}_{\mathrm{K}}) \otimes \hat{\sigma}(\mathbf{F}_{\mathrm{CFG}}) + \mathbf{D}_0 \tag{9}$$

where $\otimes$ is element-wise multiplication, $\mathbf{Conv}_3(\cdot)$ is a convolution layer with a 3×3 kernel and $\hat{\sigma}$ is a sigmoid function.

3.2.3. Cross-Attention Transformer Layer

The proposed cross-attention transformer layer (CATL) is modified from the standard MSA block presented in [13]. The two significant differences are; First, we use a cross-attention mechanism instead of self-attention. We demonstrate the effectiveness of using a cross-attention mechanism in Section 4.4. Second, cross-attention is computed locally for each window, ensuring linear complexity with image size, which makes it feasible for large inputs to be handled, as is often the case in SR.

Given as input feature map $\mathbf{F} \in \mathbb{R}^{\hat{H}\times\hat{W}\times d}$ extracted from either color or depth images, we first construct $\mathbf{F_{win}} \in \mathbb{R}^{\hat{H}\hat{W}/M^2\times M^2\times d}$ by partitioning $\mathbf{F}$ into $M \times M$ non-overlapping windows. Zero padding is applied during the partitioning process if necessary. Similarly to [55], relative position embeddings are added to $\mathbf{F_{win}}$ so that positional information can be retained. In a similar manner, the process is performed for both color and depth feature maps; we refer to this joint embedding as $\mathbf{Z}_I^0$ and $\mathbf{Z}_D^0$ for the color and depth, respectively.

In each CATL, the MSA module is replaced with a windows-based cross-attention MSA ($\mathrm{MSA_{ca}}$), while the other layers remain unchanged. By applying Equation (2) locally within each $M \times M$ window, we avoid global attention computations. Moreover, keys and values are calculated from the depth feature map, while the queries are calculated from the color feature map. Specifically, as illustrated in Figure 2b, our modified CATL consists of $\mathrm{MSA_{ca}}$ followed by a 2-layer MLP with GELU nonlinearity. Every MLP and $\mathrm{MSA_{ca}}$ module is preceded by an LN layer, and each module is followed by a residual connection.

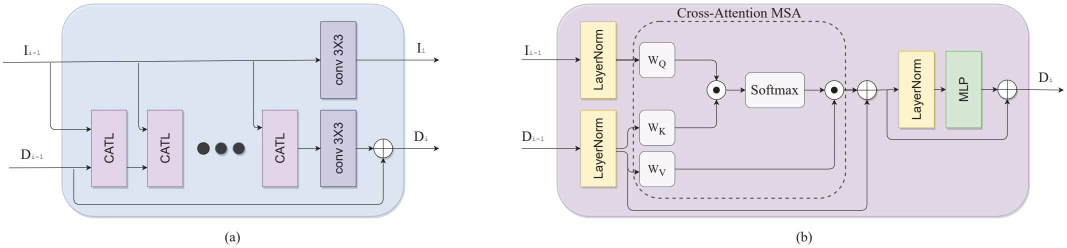


Figure 2. (**a**) Cross-Attention Transformer Block. (**b**) Cross-Attention Transformer Layer.

To enable cross-window connections in consecutive layers, regular and shifted window partitionings are used alternately. In shifted window partitioning, features are shifted by $M/2, M/2$ pixels. Finally, the CATL can be formalized as

$$\hat{\mathbf{Z}} = \mathbf{MSA_{ca}}(\mathbf{LN}(\mathbf{Z}_I^0, \mathbf{Z}_D^0)) + \mathbf{Z}_D^0 \tag{10}$$

$$\mathbf{Z} = \mathbf{MLP}(\mathbf{LN}(\hat{\mathbf{Z}}^1)) + \hat{\mathbf{Z}}^1 \tag{11}$$

where $\hat{\mathbf{Z}}$ and $\mathbf{Z}$ denote the output features of the $\mathrm{MSA}_{\mathrm{ca}}$ and MLP modules, respectively.

3.2.4. Upsampling Module

The upsampling module operates on the CAGM output, scaled via the CFG module, as elaborated in Section 3.2.2. It aims to recover high-frequency details and reconstruct the HR depth successfully. The CAGM output is first passed through a convolution layer followed by a ReLU activation function to aggregate shallow and deep features from the CAGM. Then, we use a pixel shuffle module [56] to upsample the feature map to the HR resolution. Each pixel shuffle module can perform upsampling by a factor of two or three, and we concatenate such modules according to the desired scaling factor. Finally, the upsampled feature maps are passed through another convolution layer that outputs the reconstructed depth. The parameters of the entire upsampling module are learned in the training process to improve model representation.

Formally, given the output of the CAGM module $\mathbf{F}_{\mathrm{CAGM}} \in \mathbb{R}^{H/s \times W/s}$, where s is the scaling factor, the upsampling module performs an upsampling by a factor s to reconstruct $\mathbf{D}_{\mathrm{HR}} \in \mathbb{R}^{H \times W}$. The upsampling process for a given s can be formulated as follows:

$$\begin{aligned}
&\mathbf{F}_{\mathrm{US},0} = \mathbf{Conv}_3(\mathbf{F}_{\mathrm{CAGM}}) \\
&\mathbf{F}_{\mathrm{US},i} = \mathrm{PixellShuffle}_i(\mathbf{F}_{\mathrm{US},i-1}), \quad i = 1 \ldots \log_2(s) \\
&\mathbf{D}_{\mathrm{HR}} = \mathbf{Conv}_3(\mathbf{F}_{\mathrm{US},i}).
\end{aligned} \tag{12}$$

where $\mathbf{Conv}_3(\cdot)$ is a convolution layer with a 3×3 kernel. More implementation details are given in Section 4.1.

4. Experiments

4.1. Training Details

We constructed train and test data similarly to [3,4,23,25] using 92 pairs of depth and color images from the MPI Sintel depth dataset [57] and the Middlebury depth dataset [58–60]. The training and validation pairs used in this study are similar to the ones used in [4,23]. We refer the reader to [57,58] for further information on the data included in the aforementioned datasets.

During training, we randomly extracted patches from the full-resolution images and used these as input to the network. We used an LR patch size of 96 × 96 pixels to reduce memory requirements and computation time since using larger patches had no significant impact on training accuracy. Consequently, we used HR patches of 192 × 192 and 384 × 384 for upsampling factors of 2 and 4, respectively. Given that some full-scale images had a full resolution of < 400, we used LR patch sizes of 48 × 48 and 24 × 24 for upsampling factors 8 and 16, respectively. In order to generate the LR patches, each HR patch was downsampled with bicubic interpolation. As an augmentation method, we used a random horizontal flip while training.

4.2. Implementation Details

We construct the CAGM module in the proposed architecture by stacking $K = 6$ CATBs. Each CATB is constructed from $L = 6$ CATL modules as described in Section 3.2.2. These values for K and L provided the best performance to network size trade-off in the experiments, and Section 4.4, we report results with other configurations. All convolution layers have a stride of one with zero padding, so the features' size remains fixed. Throughout the network, in convolution and transformer blocks, we use a feature (embedding) dimension size of $d = 64$. We output depth values from the final convolution layer, which has only one filter. For window partitioning in the CATL, we use $M = 12$, and each MSA module has six attention heads.

We used the PyTorch framework [61] to train a dedicated network for each upsampling factor $s \in 2, 4, 8, 16$. Each network was trained for 3×10^5 iterations and optimized using the $\mathcal{L}_1$ loss and the ADAM optimizer [62] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We used a learning rate of 10^{-4}, dividing the learning rate by 2 for every 1×10^5 iteration. All the models were trained on a PC with an i9-10940x CPU, 128GB RAM, and two Quadro RTX6000 GPUs.

4.3. Results

This section provides quantitative and qualitative evaluations of the proposed architecture for guided DSR. Our proposed architecture was evaluated on both the noise-free and the noisy Middlebury datasets. Further, we conduct experiments on the NYU Depth v2 dataset in order to demonstrate the generalization capabilities of the proposed architecture. We compare the results to other state-of-the-art methods, including global optimization-based methods [10,32], a sparse representation-based method [14], and mainly state-of-the-art deep learning-based methods [3,4,7,17,19–23,25,37,39,43,44]. We also report the results of naive bicubic interpolation as a baseline.

4.3.1. Noise-Free Middlebury Dataset

The Middlebury dataset provides high-quality depth and color image pairs from several challenging scenes. First, we evaluate the different methods for the noise-free Middlebury RGB-D datasets for different scaling factors. In Table 1, we report the obtained RMSE values. Boldface indicates the best RMSE for each evaluation, while the underline indicates the second best. In Table 1, all results are calculated from upsampled depth maps provided by the authors or generated by their code.

Clearly, from Table 1 we conclude that deep learning-based methods [3,4,7,17,19–23,25,37] outperform the more classic methods for DSR. In terms of RMSE values, the proposed architecture provides the best performance across almost all scaling factors. For large scaling factors, e.g., $8, 16$, which are difficult for most methods, our method provides good reconstruction with the lowest RMSE error across all datasets. For scaling factors x4/x8/x16, our method obtained 0.48/0.99/1.55 as the average RMSE for the entire test set, respectively. Our results outperform the second-best results in terms of average RMSE values by 0.01/0.09/0.16, respectively.

In Figures 3 and 4, we provide upsampled depth maps on the "Art" and "Moebius" datasets and a scale factor of 8 for qualitative evaluation. Upsampled depth maps are generated from 5 state-of-the-art methods, which are MSG [4], DSR [3], RDGE [32], RDN [7] and CTG [23]. We also provide bicubic interpolation as a baseline for comparison. Compared with competing methods, the proposed architecture provides more detailed HR depth boundaries. Additionally, our approach mitigates the texture-copying effect evident in some other methods, as shown by the red arrow. A significant factor contributing to these results is the attention mechanism built into the transformer model. This attention mechanism transfers HR information from the guidance image to the upsampling process in a sophisticated manner. Moreover, the transformer's ability to consider both local and global information is key to improved performance at large scaling factors. Finally, these evaluations indicate that our CAGM contributes significantly to the success of depth map SR and enables accurate reconstruction even in complex scenarios with various degradations.

Table 1. An analysis of RMSE Values for different scaling factors on the noise-free Middlebury dataset. Boldface indicates the best RMSE for each evaluation, while the underline indicates the second best.

Method	**Art**			**Books**			**Laundry**			**Dolls**			**Moebius**			**Reindeer**		
	x4	**x8**	**x16**	**x4**	**x8**	**x16**	**x4**	**x8**	**x16**	**x4**	**x8**	**x16**	**x4**	**x8**	**x16**	**x4**	**x8**	**x16**
Bicubic	3.88	5.60	8.58	1.56	2.24	3.36	2.11	3.10	4.47	1.21	1.78	2.57	1.40	2.05	2.95	2.51	3.92	5.72
TGV [10]	4.06	5.08	7.61	2.21	2.47	3.54	2.20	3.92	6.75	1.42	2.05	4.44	2.03	2.58	3.50	2.67	4.29	8.80
JID [14]	1.92	2.76	5.74	0.71	1.01	1.93	1.10	1.83	3.62	0.92	1.26	1.74	0.89	1.27	2.13	1.41	2.12	4.64
RDGE [32]	3.26	4.31	6.78	1.53	2.18	2.92	2.06	2.87	4.22	1.49	1.94	2.45	1.44	2.21	2.79	2.58	3.24	4.90
MSG [4]	1.49	2.79	5.95	0.66	1.09	1.87	1.02	1.35	2.03	0.72	0.99	1.59	0.68	1.14	2.07	1.33	1.72	2.99
DSR [3]	1.21	2.23	3.95	0.60	0.89	1.51	0.75	1.21	1.89	0.81	1.10	1.60	0.67	0.96	1.57	0.96	1.57	2.54
PSR [25]	1.59	2.57	4.83	0.83	1.19	1.70	0.92	1.52	2.97	0.91	1.31	1.88	0.86	1.21	1.87	1.11	1.80	3.11
MFR [37]	1.54	2.71	4.35	0.63	1.05	1.78	1.11	1.75	3.01	0.89	1.22	1.74	0.72	1.10	1.73	1.23	2.06	3.74
RDN [7]	1.47	2.60	4.16	0.62	1.00	1.68	0.96	1.63	2.86	0.88	1.21	1.71	0.69	1.06	1.65	1.17	1.60	3.58
PMBA [21]	1.19	2.47	4.37	0.53	1.10	1.51	0.80	1.54	2.72	0.66	1.08	1.75	0.55	1.13	1.62	0.92	1.76	2.86
RYN [20]	0.98	2.04	3.37	0.36	0.73	1.37	0.64	1.21	2.01	0.59	0.97	**1.37**	0.50	0.81	1.37	0.74	1.41	2.22
CUN [22]	1.05	2.27	3.67	0.35	0.73	1.45	0.59	1.15	2.25	0.61	0.97	1.43	0.48	0.77	1.31	0.82	1.51	2.38
GDC [19]	1.09	2.04	3.58	0.38	0.68	1.41	0.64	1.13	2.13	0.63	0.97	1.44	0.49	0.79	1.37	0.84	1.51	2.43
TDTN [39]	1.24	2.45	–	0.48	0.86	–	0.68	1.29	–	0.76	1.15	–	0.61	0.91	–	0.95	1.75	–
MIG [43]	1.46	2.74	4.26	0.58	0.95	1.67	0.93	1.57	2.85	0.87	1.21	1.75	0.66	1.04	1.66	1.17	2.11	3.81
PDR [44]	1.59	2.57	4.83	0.83	1.19	1.70	0.92	1.52	2.97	0.91	1.31	1.88	0.86	1.21	1.87	1.11	1.80	3.11
CTG [23]	0.73	1.89	2.76	0.35	**0.66**	1.22	**0.43**	0.87	1.62	0.50	0.90	1.49	**0.46**	0.76	1.31	**0.43**	1.19	1.84
FCTN (Proposed)	**0.71**	**1.71**	**2.56**	**0.34**	0.68	**1.12**	0.47	**0.79**	**1.43**	**0.45**	**0.81**	1.40	**0.46**	**0.68**	**1.18**	0.47	**1.12**	**1.64**

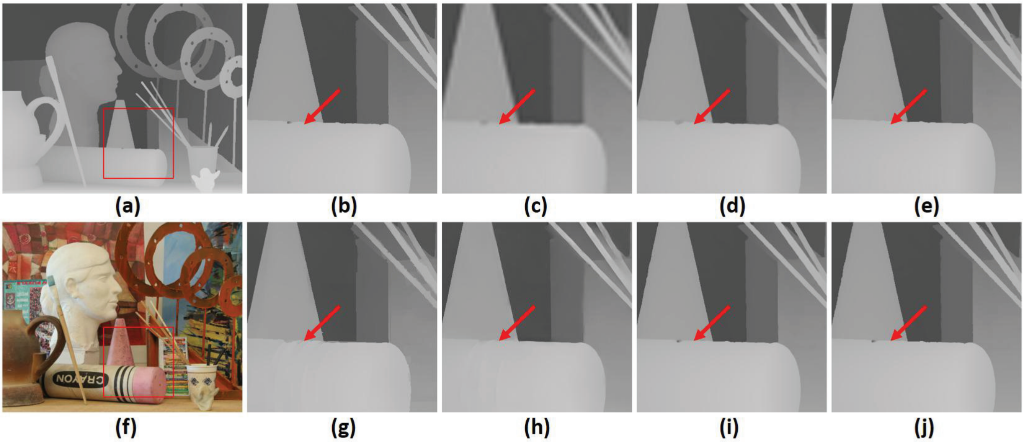


Figure 3. A visual quality comparison for depth map SR at a scale factor of 8 on the noise-free "art" dataset. (**a**) HR depth image, (**f**) HR color image, (**b**) extracted ground truth patch (marked by a red square), and upsampled patches by (**c**) Bicubic, (**d**) MSG [4], (**e**) DSR [3], (**g**) RDGE [32], (**h**) RDN [7], (**i**) CTG [23], (**j**) the proposed FCTN method (best viewed on the enlarged electronic version).

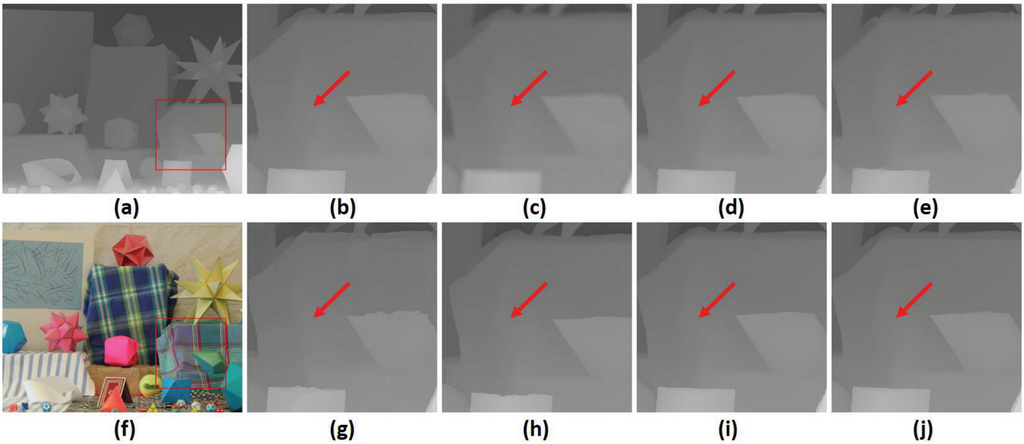


Figure 4. A visual quality comparison for depth map SR at a scale factor of 8 on the noise-free "Moebius" dataset. (**a**) HR depth image, (**f**) HR color image, (**b**) extracted ground truth patch (marked by a red square, and upsampled patches by (**c**) Bicubic, (**d**) MSG [4], (**e**) DSR [3], (**g**) RDGE [32], (**h**) RDN [7], (**i**) CTG [23], (**j**) the proposed FCTN method (best viewed on the enlarged electronic version).

4.3.2. Noisy Middlebury Dataset

We further demonstrate the robustness of the proposed architecture on the noisy Middlebury dataset. We added Gaussian noise to the LR training data, simulating the case where depth maps are corrupted during acquisition, in the same way as [3,7,23,37]. All the models were retrained and evaluated on a test set corrupted with the same Gaussian noise. For the noisy dataset, we report the RMSE values in Table 2.

Table 2. An analysis of RMSE values for different scaling factors on the noisy Middlebury dataset. Boldface indicates the best RMSE for each evaluation, while the underline indicates the second best.

Method	**Art**		**Books**		**Laundry**		**Dolls**		**Moebius**		**Reindeer**	
	x8	**x16**	**x8**	**x16**	**x8**	**x16**	**x8**	**x16**	**x8**	**x16**	**x8**	**x16**
Bicubic	6.74	9.04	4.68	5.30	5.35	6.53	4.51	4.90	4.54	5.02	5.71	7.12
TGV [10]	7.26	12.05	2.88	4.73	4.45	8.06	2.82	5.14	3.01	6.11	4.65	9.03
MSG [4]	4.24	7.42	2.48	4.19	3.31	4.88	2.53	3.41	2.47	3.76	3.36	4.95
MFR [37]	3.97	6.14	2.13	3.17	2.82	4.57	2.25	3.30	2.13	3.33	3.01	4.86
RDN [7]	4.09	6.62	2.11	3.36	2.88	5.11	2.33	3.59	2.18	3.69	3.09	4.93
DSR [3]	–	6.96	–	5.66	–	7.54	–	4.28	–	3.39	–	5.25
RYN [20]	3.47	–	1.88	–	2.47	–	1.97	–	1.87	–	2.68	–
GDC [19]	3.31	4.77	1.69	**2.46**	2.20	3.36	1.89	2.59	1.72	2.68	2.57	3.44
JIIF [40]	3.87	7.14	1.75	2.47	–	–	–	–	2.03	3.18	–	–
MIG [43]	3.95	6.15	2.10	3.17	3.00	4.88	2.21	3.51	2.12	3.51	3.04	4.97
CTG [23]	3.26	4.72	1.61	2.96	1.63	3.47	1.64	**2.16**	1.63	2.24	**1.79**	3.59
FCTN (Proposed)	**3.01**	**4.55**	**1.54**	2.66	**1.61**	3.15	**1.59**	2.32	**1.27**	**2.09**	1.81	**3.17**

Our first observation is that noise added to the LR depth maps significantly affects the reconstructed HR depth maps regardless of the method or scaling factor used. However, the proposed architecture still generates clean and sharp reconstructions and outperforms competing methods in terms of RMSE.

An even more realistic scenario is that data acquired by both the depth and color sensors are corrupted by noise. Our method was further tested by adding Gaussian noise with a mean of 0 and a standard deviation of 5 to the HR guidance images. This was done both in training and in testing. We again retrained the models and report the obtained average RMSE values in Table 3. In Table 3, we observe that the added noise in the HR guidance image did not significantly affect the performance of our method, compared to only adding noise to LR depth. According to our results, the proposed CAGM is somewhat insensitive to noise added to the guidance image.

Table 3. An Analysis of the average RMSE values for different noise schemes.

Middlebury Dataset Version	x4	x8	x16
Noise-Free	0.48	0.99	1.55
Depth Noise	1.17	1.80	2.99
Depth and Color Noise	1.35	2.01	3.19

4.3.3. NYU Depth v2 Dataset

In this section, the proposed architecture is tested on the challenging public NYU Depth v2 [63] dataset as a means of demonstrating its generalization ability. There are 1449 high-quality RGB-D images of natural indoor scenes in this dataset, with apparent misalignments between depth maps and color images. We note that data from NYU Depth v2 are very different from the Middlebury Dataset and were not included in the training data of our models.

We report the average RMSE value across the entire dataset in Table 4. Boldface indicates the best RMSE value. As a baseline, we report the results of Bicubic interpolation as well as the results of competing guided SR approaches; ATGV-Net [5], MSG [4], DSR [3], RDN [7], RYN [20], PMBA [21], DEAF [42], JIIF [40], DCT [17], and CTG [23]. The proposed architecture achieves the lowest average RMSE, demonstrating the proposed method's generalization ability and robustness.

Table 4. Quantitative comparisons of the ablation experiments. Reported results are average RMSE on the noise-free Middlebury dataset for scaling factors 4, 8, and 16. Boldface indicates the best RMSE for each evaluation, while the underline indicates the second best.

Method	Average RMSE on NYU Depth v2 Dataset
Bicubic	2.36
ATGV-Net [5]	1.28
MSG [4]	1.31
RDN [7]	1.21
DSR [3]	1.34
RYN [20]	1.06
PMBA [21]	1.06
DEAF [42]	1.12
JIIF [40]	1.37
DCT [17]	1.59
CTG [23]	<u>0.95</u>
FCTN (Proposed)	**0.91**

4.3.4. Inference Time

For a DSR method to applyto real-world applications, it is often required to work in a close-to-real-time performance. Thus, we report the inference time of the proposed architecture compared to other competing approaches. Inference times were measured using an image of size 1320 × 1080 pixels and the setup described in Section 4.2. We report our results in milliseconds in Table 5

Table 5 shows that compared to traditional methods, the proposed architecture, as well as other deep learning-based methods, provide significantly faster inference times. Moreover, the proposed method is comparable to competing methods and achieves lower RMSE values. In contrast, References [10,12,32] require multiple optimization iterations to obtain accurate reconstructions, leading to slower inference times. Some methods, such as [3,32], upsample the LR depth as an initial preprocessing step before the image is fed to the model. As a result, they show very similar inference times regardless of the scaling factor.

Table 5. Average inference times (milliseconds) for different scaling factors.

Method	x2	x4	x8	x16
Bicubic	10	10	10	10
TGV [10]	45,730	49,780	46,340	46,200
AR [12]	158,010	157,730	157,950	158,770
RDGE [32]	68,070	67,690	68,450	68,170
MSG [4]	260	300	380	420
DSR [3]	220	230	230	230
RYN [20]	460	630	720	880
CTG [23]	150	380	480	530
FCTN (Proposed) [23]	140	304	420	490

4.4. Ablation Study

In the ablation study, we test the effects of the CATB number in the CAGM and CATL number in each CATB on model performance. Results are shown in Figure 5a,b, respectively. It is observed that the RMSE of the reconstructed depth is positively correlated with both hyperparameters until it becomes eventually saturated. As we increase either hyperparameter, model size becomes increasingly prominent, and training\inference time and memory requirements are negatively impacted. Thus, to balance the performance and model size, we choose 6 for both hyperparameters as described in Section 4.2. CATL numbers were evaluated with a configuration of $K = 6$ CATBs.

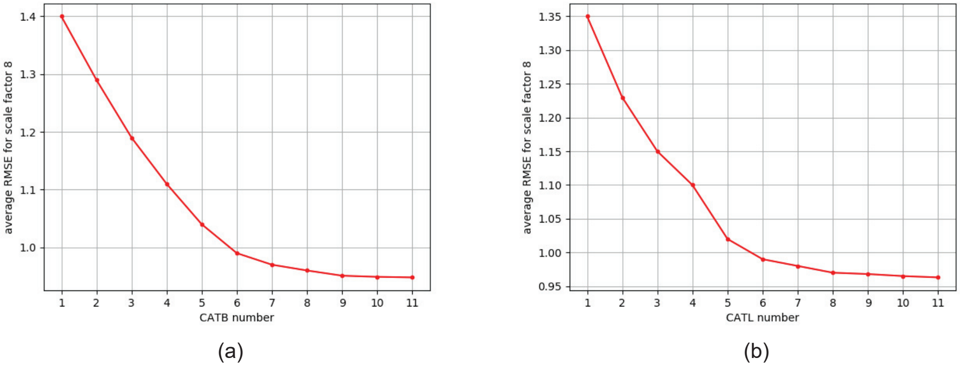


Figure 5. Ablation study on different configurations of the proposed CAGM. Results are the average RMSE on the noise-free Middlebury dataset for scaling factor 8. (**a**) The effect of the CATB number in the CAGM, and (**b**) the effect of CATL number in each CATB.

The impact of each component in our design is evaluated via the following experiments: (1) Our architecture without any guidance from the color image, denoted as "Depth-Only". (2) Our architecture without shifted windows in the CATL, denoted "w/o shift”. (3) Our architecture without the CFG module, denoted “w/o CFG”. (4) Our architecture without the use of cross-attention for guidance. In this setting, we replaced the CATL with a similar design using only self-attention with depth features as input. Features from the color image were concatenated after every modified CATL to provide guidance. We denote this setting as “w/o cross-attention”.

We evaluate the different designs on the Middlebury test set at scaling factors 4, 8, and 16. We use the same CATB and CATL configuration as described in Section 4.2 in these experiments. We summarize the results in Table 6 and observe that: (1) As expected, using only the LR depth for DSR without guidance from a color image provides inferior results. (2) As also observed in [47], incorporating shifted window partitioning into our CATL improves the performance. Using shifted windows partitioning enables connections among windows in the preceding layers, improving the representation capability of each CATL. (3) Our CFG module provides additional high-frequency information directly to the upsampling module. As a result, the upsampling module can reconstruct a higher quality HR depth, and we observe that performance improves slightly. (4) We observe that using a simple concatenation of features instead of the proposed cross-attention guidance leads to inferior results. Incorporating the guidance from the color image via cross-attention allows the color feature to interact elaborately with the depth features and to encode long-distant dependencies between the two modalities.

Table 6. An analysis of the average RMSE values for different ablation experiments on the noise-free Middlebury dataset. Boldface indicates the best RMSE for each evaluation.

Design	**Depth-Only**			**w/o Shift**			**w/o CFG**			**w/o Cross-Attention**			**FCTN (Proposed)**		
Scale Factor	x4	x8	x16	x4	x8	x16	x4	x8	x16	x4	x8	x16	x4	x8	x16
RMSE	0.65	1.39	3.01	0.52	1.14	1.90	0.51	1.06	1.79	0.59	1.28	2.17	**0.48**	**0.99**	**1.55**

5. Conclusions

We introduce a novel transformer-based architecture with cross-attention for guided DSR. First, a shallow feature extraction module extracts meaningful features from LR depth and HR color images. These features are fed to a cascaded transformer module with cross-attention, which extracts more elaborate features while simultaneously incorporates guidance from the color features via the cross-attention mechanism. The cascaded transformer module is constructed by stacking transformer layers with shifted window partitioning, which enables interactions between windows in consecutive layers. Using such a design, the proposed architecture achieves state-of-the-art results on the DSR benchmarks. At the same time, model size and inference time remain comparably small, making our architecture usable for real-world applications.

Our future work will explore more realistic depth artifacts (e.g., sparse depth values, misalignment between guidance and depth images, etc.). Moreover, we will examine the proposed architecture on additional real-world continuous data acquired from sensors mounted, e.g., on an autonomous robot.

Author Contributions: Conceptualization, I.A.; Methodology, I.A. and I.C.; Writing—original draft, I.A.; Writing—review & editing, I.C.; Supervision, I.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the PMRI—Peter Munk Research Institute-Technion.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: MPI Sintel Dataset—[http://sintel.is.tue.mpg.de/, accessed on 15 February 2023]. Middlebury Stereo Datasets—[https://vision.middlebury.edu/stereo/data/, accessed on 15 February 2023].

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

References

1. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
2. Schamm, T.; Strand, M.; Gumpp, T.; Kohlhaas, R.; Zollner, J.M.; Dillmann, R. Vision and ToF-based driving assistance for a personal transporter. In Proceedings of the 2009 International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–6.
3. Guo, C.; Li, C.; Guo, J.; Cong, R.; Fu, H.; Han, P. Hierarchical features driven residual learning for depth map super-resolution. *IEEE Trans. Image Process.* **2018**, *28*, 2545–2557. [CrossRef]
4. Hui, T.W.; Loy, C.C.; Tang, X. Depth map super-resolution by deep multi-scale guidance. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 353–369.
5. Riegler, G.; Rüther, M.; Bischof, H. Atgv-net: Accurate depth super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 268–284.
6. Song, X.; Dai, Y.; Qin, X. Deeply supervised depth map super-resolution as novel view synthesis. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2323–2336. [CrossRef]
7. Zuo, Y.; Fang, Y.; Yang, Y.; Shang, X.; Wang, B. Residual dense network for intensity-guided depth map enhancement. *Inf. Sci.* **2019**, *495*, 52–64. [CrossRef]
8. He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1397–1409. [CrossRef] [PubMed]
9. Yang, Q.; Yang, R.; Davis, J.; Nistér, D. Spatial-depth super resolution for range images. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 17–22 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.
10. Ferstl, D.; Reinbacher, C.; Ranftl, R.; Rüther, M.; Bischof, H. Image guided depth upsampling using anisotropic total generalized variation. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 993–1000.
11. Jiang, Z.; Hou, Y.; Yue, H.; Yang, J.; Hou, C. Depth super-resolution from RGB-D pairs with transform and spatial domain regularization. *IEEE Trans. Image Process.* **2018**, *27*, 2587–2602. [CrossRef]
12. Yang, J.; Ye, X.; Li, K.; Hou, C.; Wang, Y. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. *IEEE Trans. Image Process.* **2014**, *23*, 3443–3458. [CrossRef]
13. 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; pp. 5998–6008.
14. Kiechle, M.; Hawe, S.; Kleinsteuber, M. A joint intensity and depth co-sparse analysis model for depth map super-resolution. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1545–1552.
15. Kwon, H.; Tai, Y.W.; Lin, S. Data-driven depth map refinement via multi-scale sparse representation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 159–167.
16. Park, J.; Kim, H.; Tai, Y.W.; Brown, M.S.; Kweon, I. High quality depth map upsampling for 3d-tof cameras. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1623–1630.
17. Zhao, Z.; Zhang, J.; Xu, S.; Lin, Z.; Pfister, H. Discrete cosine transform network for guided depth map super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5697–5707.
18. Lutio, R.d.; D'aronco, S.; Wegner, J.D.; Schindler, K. Guided super-resolution as pixel-to-pixel transformation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8829–8837.
19. Kim, J.Y.; Ji, S.; Baek, S.J.; Jung, S.W.; Ko, S.J. Depth Map Super-Resolution Using Guided Deformable Convolution. *IEEE Access* **2021**, *9*, 66626–66635. [CrossRef]
20. Li, T.; Dong, X.; Lin, H. Guided depth map super-resolution using recumbent y network. *IEEE Access* **2020**, *8*, 122695–122708. [CrossRef]
21. Ye, X.; Sun, B.; Wang, Z.; Yang, J.; Xu, R.; Li, H.; Li, B. Pmbanet: Progressive multi-branch aggregation network for scene depth super-resolution. *IEEE Trans. Image Process.* **2020**, *29*, 7427–7442. [CrossRef]
22. Cui, Y.; Liao, Q.; Yang, W.; Xue, J.H. RGB Guided Depth Map Super-Resolution with Coupled U-Net. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
23. Ariav, I.; Cohen, I. Depth Map Super-Resolution via Cascaded Transformers Guidance. *Front. Signal Process.* **2022**, *3*.. [CrossRef]
24. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3929–3938.
25. Huang, L.; Zhang, J.; Zuo, Y.; Wu, Q. Pyramid-Structured Depth Map Super-Resolution Based on Deep Dense-Residual Network. *IEEE Signal Process. Lett.* **2019**, *26*, 1723–1727. [CrossRef]

26. He, K.; Sun, J.; Tang, X. Guided image filtering. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–14.
27. Liu, M.Y.; Tuzel, O.; Taguchi, Y. Joint geodesic upsampling of depth images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 169–176.
28. Lu, J.; Forsyth, D. Sparse depth super resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2245–2253.
29. Dong, W.; Shi, G.; Li, X.; Peng, K.; Wu, J.; Guo, Z. Color-guided depth recovery via joint local structural and nonlocal low-rank regularization. *IEEE Trans. Multimed.* **2016**, *19*, 293–301. [CrossRef]
30. Ham, B.; Cho, M.; Ponce, J. Robust image filtering using joint static and dynamic guidance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4823–4831.
31. Ham, B.; Min, D.; Sohn, K. Depth superresolution by transduction. *IEEE Trans. Image Process.* **2015**, *24*, 1524–1535. [CrossRef] [PubMed]
32. Liu, W.; Chen, X.; Yang, J.; Wu, Q. Robust color guided depth map restoration. *IEEE Trans. Image Process.* **2016**, *26*, 315–327. [CrossRef] [PubMed]
33. Park, J.; Kim, H.; Tai, Y.W.; Brown, M.S.; Kweon, I.S. High-quality depth map upsampling and completion for RGB-D cameras. *IEEE Trans. Image Process.* **2014**, *23*, 5559–5572. [CrossRef]
34. Yang, J.; Ye, X.; Li, K.; Hou, C. Depth recovery using an adaptive color-guided auto-regressive model. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 158–171.
35. Riegler, G.; Ferstl, D.; Rüther, M.; Bischof, H. A deep primal-dual network for guided depth super-resolution. *arXiv* **2016**, arXiv:1607.08569.
36. Zhou, W.; Li, X.; Reynolds, D. Guided deep network for depth map super-resolution: How much can color help? In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1457–1461.
37. Zuo, Y.; Wu, Q.; Fang, Y.; An, P.; Huang, L.; Chen, Z. Multi-scale frequency reconstruction for guided depth map super-resolution via deep residual network. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 297–306. [CrossRef]
38. de Lutio, R.; Becker, A.; D'Aronco, S.; Russo, S.; Wegner, J.D.; Schindler, K. Learning Graph Regularisation for Guided Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1979–1988.
39. Yao, C.; Zhang, S.; Yang, M.; Liu, M.; Qi, J. Depth super-resolution by texture-depth transformer. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
40. Tang, J.; Chen, X.; Zeng, G. Joint implicit image function for guided depth super-resolution. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, China, 20–24 October 2021; pp. 4390–4399.
41. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
42. Liu, P.; Zhang, Z.; Meng, Z.; Gao, N. Deformable Enhancement and Adaptive Fusion for Depth Map Super-Resolution. *IEEE Signal Process. Lett.* **2021**, *29*, 204–208. [CrossRef]
43. Zuo, Y.; Wang, H.; Fang, Y.; Huang, X.; Shang, X.; Wu, Q. MIG-net: Multi-scale Network Alternatively Guided by Intensity and Gradient Features for Depth Map Super-resolution. *IEEE Trans. Multimed.* **2021**, *24*, 3506–3519. [CrossRef]
44. Liu, P.; Zhang, Z.; Meng, Z.; Gao, N.; Wang, C. PDR-Net: Progressive depth reconstruction network for color guided depth map super-resolution. *Neurocomputing* **2022**, *479*, 75–88. [CrossRef]
45. 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.
46. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv* **2021**, arXiv:2102.12122.
47. 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.
48. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Nashville, TN, USA, 19–25 June 2021; pp. 6881–6890.
49. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
50. 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; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
51. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep back-projection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1664–1673.

52. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2472–2481.
53. Xiao, T.; Singh, M.; Mintun, E.; Darrell, T.; Dollár, P.; Girshick, R. Early convolutions help transformers see better. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30392–30400.
54. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
55. Hu, H.; Zhang, Z.; Xie, Z.; Lin, S. Local relation networks for image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3464–3473.
56. Shi, W.; Caballero, J.; Huszár, 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
57. Butler, D.J.; Wulff, J.; Stanley, G.B.; Black, M.J. A naturalistic open source movie for optical flow evaluation. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 611–625.
58. Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; Westling, P. High-resolution stereo datasets with subpixel-accurate ground truth. In Proceedings of the German Conference on Pattern Recognition, Munster, Germany, 2–5 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 31–42.
59. Scharstein, D.; Pal, C. Learning conditional random fields for stereo. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 17–22 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.
60. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [CrossRef]
61. 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 Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8024–8035.
62. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
63. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 746–760.

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

Kernel Estimation Using Total Variation Guided GAN for Image Super-Resolution

Jongeun Park, Hansol Kim and Moon Gi Kang *

School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Republic of Korea
* Correspondence: mkang@yonsei.ac.kr

Abstract: Various super-resolution (SR) kernels in the degradation model deteriorate the performance of the SR algorithms, showing unpleasant artifacts in the output images. Hence, SR kernel estimation has been studied to improve the SR performance in several ways for more than a decade. In particular, a conventional research named KernelGAN has recently been proposed. To estimate the SR kernel from a single image, KernelGAN introduces generative adversarial networks(GANs) that utilize the recurrence of similar structures across scales. Subsequently, an enhanced version of KernelGAN, named E-KernelGAN, was proposed to consider image sharpness and edge thickness. Although it is stable compared to the earlier method, it still encounters challenges in estimating sizable and anisotropic kernels because the structural information of an input image is not sufficiently considered. In this paper, we propose a kernel estimation algorithm called Total Variation Guided KernelGAN (TVG-KernelGAN), which efficiently enables networks to focus on the structural information of an input image. The experimental results show that the proposed algorithm accurately and stably estimates kernels, particularly sizable and anisotropic kernels, both qualitatively and quantitatively. In addition, we compared the results of the non-blind SR methods, using SR kernel estimation techniques. The results indicate that the performance of the SR algorithms was improved using our proposed method.

Keywords: kernel estimation; generative adversarial networks; super-resolution; self-similarity; total variation; KernelGAN; structural information

Citation: Park, J.; Kim, H.; Kang, M.G. Kernel Estimation Using Total Variation Guided GAN for Image Super-Resolution. *Sensors* **2023**, *23*, 3734. https://doi.org/10.3390/s23073734

Academic Editor: Benoit Vozel

Received: 28 February 2023
Revised: 21 March 2023
Accepted: 3 April 2023
Published: 4 April 2023

1. Introduction

High-resolution (HR) images are required in various applications, for example, medical or satellite imaging, wherein specific objects must be distinguished or patterns must be recognized. However, the observed images often have low resolution (LR) because of the physical limitation of the small image sensor or the image acquisition environments. Single image super-resolution (SISR) algorithms for recovering HR images from LR images, have been extensively studied for decades. By overcoming the limitations of the observed LR image, the desired information can be exploited, or the hardware cost efficiency can be achieved. The LR image observation model, also referred to as the degradation model, is described as follows [1]:

$$y = DBMx + n, \tag{1}$$

where y represents the LR image, x represents the HR image, DBM is the degradation operation comprising the downsampling matrix D, blurring matrix (blurring kernel, SR kernel, point spread function; PSF) B, and warping matrix M while n represents the additive white noise. Image super-resolution (SR) reconstruction is generally a severely ill-posed problem because the information from an LR image is usually insufficient and the blurring matrix B is typically unknown.

To overcome above inherent physical limitations and obtain an accurate HR image, numerous methods have been proposed in two branches: (i) classical approaches [2–8], and (ii) deep-learning-based approaches [9–16]. In the classical SISR, studies have generally focused on addressing the ill-posedness resulting from insufficient information in the LR image and inaccurate registration by using methods based on regularizing the image prior [2–5] or exploiting the recurrence property of the internal image patches [6–8]. However, the blurring matrix B in these methods is usually assumed to be known from measurements or simple blurring such as a Gaussian kernel or bicubic kernel. Early deep-learning-based approaches [9–15], HR images were degraded using a Gaussian or bicubic kernel to generate LR-HR dataset pairs. However, this dataset generation method is insufficient for representing natural LR images because the blurring matrix B varies depending on the image acquisition environment. Because only a single image is given in SISR, the information that can be used for the SR is limited to B or the image priors. A comparison between the SISR results using the assumed kernel and the estimated kernel is shown in Figure 1. The SISR result using the assumed kernel in Figure 1b shows a blurry result without any resolution improvement. However, when the estimated kernel is used, the resolution of the SISR result is improved, evident through the clear visibility of whiskers and patterns of fur in Figure 1c. Therefore, the blurring kernels in the SR process have to be considered to improve the performance of the SISR algorithms.

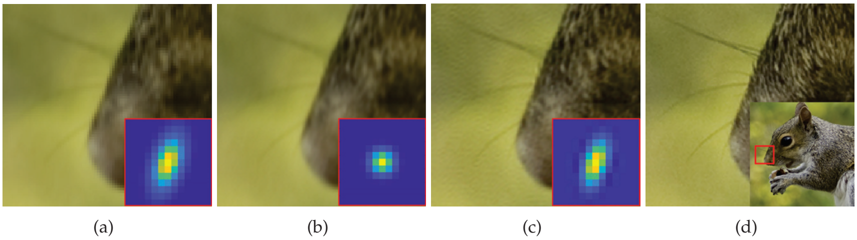


Figure 1. Comparison of SISR results for scale factor of ×2 using [13]. (**a**) input LR image degraded with ground truth(GT) blurring kernel. (**b**) SISR result with a kernel assumed as Gaussian kernel. (**c**) SISR result with an estimated kernel. (**d**) GT image.

A multitude of methods has been proposed to address this issue in real-world SISR [17–19]. Ji et al. [16] proposed a method inspired by KernelGAN [20] that constructs a kernel pool from a high-quality source image using kernel estimation techniques before generating an LR image through degradation. This demonstrates that the degradation process, particularly the blurring process, can be effectively modeled using kernel estimation methods. Despite its advantages, KernelGAN may exhibit inconsistencies or instabilities owing to the inherent randomness of GAN. Liang et al. [21] introduced a kernel-pool generation method, flow-based kernel prior (FKP), which exploits invertible mapping between a random variable and a kernel using several flow blocks. It achieved stable kernel estimation performance. However, their method required pre-training and could not estimate an accurate kernel if the desired kernel was not included in the kernel pool. Kim et al. [22] proposed an enhanced version of KernelGAN that exploits the distinctive properties of LR-HR image pairs. Their method demonstrated improved performance compared to KernelGAN, but still encountered challenges in estimating sizable and anisotropic kernels.

For this study, we proposed a kernel estimation method that addresses the challenge of accurately estimating sizable and anisotropic kernels. The proposed method is guided by a total variation map, which emphasizes the edge regions of the image where detailed information is most prevalent, and exploits self-similarity to a greater extent than previous methods. The main contributions of the study are summarized as follows:

- The proposed method adopts a total variation map and uses it as a guide for the network to focus on the structural information of the image.
- Compared to previous methods, the proposed method is cost- and memory-efficient.
- We demonstrate that the proposed method exhibits superior performance, particularly in accurately estimating sizable and anisotropic kernels, compared to conventional methods.

The remainder of the paper is organized as follows: In Section 2, a summary of the relevant background work is provided. The proposed method is described in detail in Section 3. The experimental results are presented in Section 4, and the conclusions are presented in Section 5.

2. Background

As mentioned in the previous section, the blurring matrix B is generally assumed to be a Gaussian or a bicubic kernel in various SISR studies. However, owing to environmental factors such as camera shaking, rapid movement, and weather conditions, the blurring kernel may not be identical even if the same imaging system is used. For SISR, accurately estimating the blurring kernel is crucial because an inaccurately assumed kernel often produces reconstructed images with ringing or blurring artifacts.

Michaeli et al. [7] proposed an SR kernel estimation method for a single image using the self-similarity property of natural images, in which similar structures are repeated across scales. In their method, patches with explicit structural similarities were matched, and the SR kernel was estimated using maximum a posteriori (MAP) optimization. KernelGAN, proposed by Bell-Kligler et al. [20], is a pioneering work that introduced a deep linear network for SR kernel estimation. Although having the same fundamental background, it employs a distinct optimization tool, GAN. In KernelGAN, the generator G generates a fake patch by downscaling a patch randomly picked from the input image, and the discriminator D determines whether it is a fake or real patch of the input image. KernelGAN is trained to create a downscaled fake patch with the same statistics as a real patch, maximizing self-similarity, such that the network reproduces the degradation process of the given input image and extracts the optimal SR kernel. KernelGAN demonstrated that the network could successfully estimate the SR kernels.

Kim et al. [22] noted that KernelGAN did not consider image sharpness and the difference in edge thickness between HR and LR images and proposed Enhanced-KernelGAN (E-KernelGAN). They consider the image's sharpness using 'degradation and ranking comparison', which indirectly utilizes the structural information of the image and improves the kernel estimation stability by excluding unsuitable candidates from the kernel space. In addition, they proposed the 'kernel correction' module as a post-processing step to refine the estimated kernel variance and resize it. This post-processing step, which considers edge thickness, also indirectly uses structural information. E-KernelGAN successfully improves the SR kernel estimation stability and accuracy, but fails to fully exploit the self-similarity property that is fundamental to SISR kernel estimation. Consequently, the estimation of sizable and anisotropic SR kernels is limited. In the next section, we propose Total Variation Guided KernelGAN (TVG-KernelGAN), which efficiently utilizes self-similarity by weighting the input image.

3. Proposed Method

3.1. Challenging Kernels and SR

Classical SR methods typically estimate HR images by solving the optimization problem as follows:

$$\hat{x} = \underset{x}{argmin}\left\{ \|y - DBMx\|_2^2 + \lambda \|\nabla x\|_p^p \right\}. \tag{2}$$

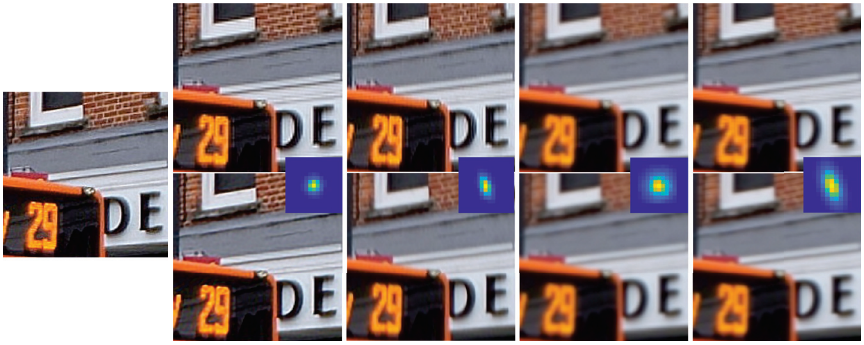
$\hat{x}$ is the optimal HR image that minimizes the given cost function. The first term is the data fidelity term, and the second term is the p-norm regularization term, which imports various image priors to suppress noise. λ is the regularization parameter that determines how much the regularization term contributes to the optimization process. In SISR, M is not considered because the given image can be located at arbitrary coordinates. D is a downsampling matrix the inverse of which is generally interpreted as an arbitrary interpolator, such as a bilinear or bicubic interpolator. The remaining factors that affect the

SR performance are B and the regularization term. Meanwhile, deep-learning-based SR methods typically use Equation (3) to predict HR images.

$$\hat{x} = F\left[\underset{\Theta}{argmin}\{\textstyle\sum_i \|F(\Theta, y_i) - x_i\|^2\}, y\right], \tag{3}$$

$$\text{where, } y_i = DBMx_i + n.$$

F is the SR network output with the network parameter Θ and input image y. Data pairs (x_i, y_i) are prepared using the degradation model in Equation (1). In general, the blurring kernel B is assumed to be bicubic or Gaussian, which limits the generalization performance of the network. To investigate the effects of different blurring kernels on the SR results, we degraded the same input image using four differently shaped kernels and applied the SR methods as shown in Figure 2. When the input image was degraded with a small round kernel, the resulting SR image showed relatively weak ringing artifacts and blurring artifacts, as shown in the first column of Figure 2b. The ringing and blurring artifacts became more severe in the second column of Figure 2b with the change to an anisotropic kernel. With a sizable kernel in the third column of Figure 2b, the SR results show severe blurring artifacts without any noticeable resolution enhancement. In the case of a sizable and anisotropic kernel in the fourth column of Figure 2b, severe blurring and ringing artifacts can be found. These results show that more sizable and anisotropic kernels severely deteriorate the SR performance; the focus of this study is this kind of kernel. As shown in Section 4, previous work on kernel estimation failed to estimate these sizable and anisotropic kernels. In this paper, we propose a method that successfully estimates these challenging kernels.



(a) (b)

Figure 2. SR results comparison using blurring kernels in scale factor ×2. (**a**) GT, (**b**) the first row shows the results of solving Equation (2) and the second row shows the results of [13]. The input images of each column are degraded, respectively, from GT using the different blurring kernels shown in the middle of each columns. However, in the SR process, both SR methods assumed the same Gaussian kernel as a blurring kernel to investigate the effects of different kernels.

3.2. Total Variation Weight Map

When a given input image is severely blurred, the network has difficulty extracting meaningful features from the given patch to distinguish between real and fake patches. Consequently, the network may converge to a meaningless local minimum. To estimate the sizable and anisotropic kernels successfully in such situations, we focused on the edges of the images. Several studies on SR and kernel estimation have used edges that contain rich structural information [15,23–25]. In particular, Cho and Lee [24] estimated an extremely directional and sharp kernel, generally known as a motion-blurring kernel, using a strong edge prior. This implies that strong edges that remain after the blurring process are still present when the same blurring process is applied. Inspired by this edge prior, we incorporated the edges of the input image into the kernel estimation process.

The proposed method aims to maximize self-similarity efficiently by weighting the edge region of the input image such that the network can focus on structural information and successfully estimates more challenging SR kernels. Because we are interested in the edge region rather than the edge itself, we require a relatively smooth weight around the edges. Farsiu et al. [2] proposed total variation using four directions, including two diagonal directions, to regularize the noise. They demonstrated that this regularization suppressed noise while preserving the edges, meaning that the total variation smoothly and gradually highlighted the edges and details.

There are several options for highlighting the edges, as shown in Figure 3. The weight maps were normalized using the maximum values of each map. Consistent with [2], Figure 3e showed the smoothest edge map with the smallest weight difference between the strong and weak edges. Therefore, we used the following four-direction total variation map:

$$\begin{aligned} \mathrm{map}_{TV} &= \underbrace{\sum_{a=0}^{1}\sum_{b=-1}^{1}}_{a+b\geq 0} ||y - P_{n1}^{a}P_{n2}^{b}y||_1, \\ w &= \mathrm{map}_{TV} + c, \end{aligned} \tag{4}$$

where P is the shift operator in the vertical direction n_1 and horizontal direction n_2, and a and b are the order of P. We focus on the edge region of the input image; however, this does not mean that the plane region has no information. Therefore, we added a constant c to map_{TV} so that the plane region is not completely discarded from the kernel estimation process.

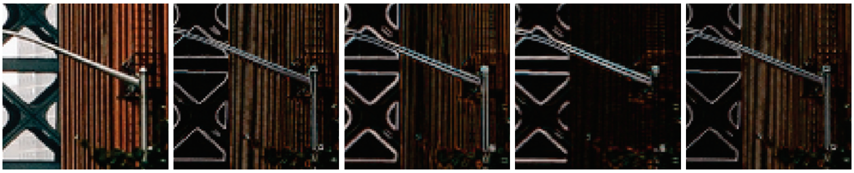


Figure 3. Normalized weights map examples, (**a**) GT, (**b**) Using forward difference for two directions and L2 norm, (**c**) Using Sobel filtering and L1 norm, (**d**) Using forward difference for two directions and L1 norm, (**e**) Using forward difference for four directions and L1 norm.

3.3. TVG-KernelGAN

The structure of the proposed method is shown in Figure 4. The input image is first weighted by the total variation map and used as an input patch for G and D. By incorporating the total variation map as a weight, the network directly utilizes the structural information of the input image and focuses on the edge region. The weighted input is not used during the entire training process but is instead used at certain iterations, i.e.,

$$\hat{y} = \begin{cases} w * y, & \text{if mod(t,s) = 0} \\ y, & \text{otherwise,} \end{cases} \tag{5}$$

where $\hat{y}$ is the input for G and D at iteration t, and s is the ratio parameter that determines the frequency of using the weight map w. In the same context as the addition of the constant c to map_{TV} in Equation (4), this switching scheme ensures that the information in the plane region is not completely discarded. Furthermore, the total variation guide scheme is applied after several tens of iterations when a general bicubic kernel shape is sufficiently formed.

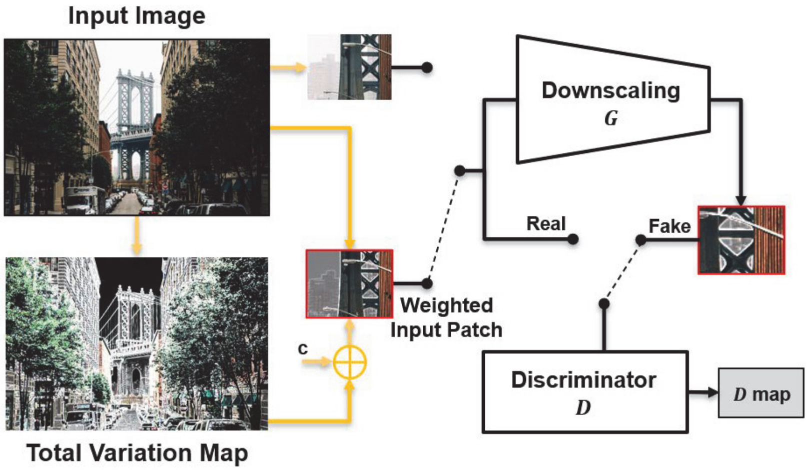


Figure 4. Structure of TVG-KernelGAN.

Finally, the TVG-KernelGAN loss function is given by

$$L_{TVG\text{-}KernelGAN} = |D(\hat{y}) - 1| + |D(G(\hat{y}))| + R_B. \tag{6}$$

Here, R_B is the kernel regularization term as in [20], and R_B is given as follows:

$$R_B = \alpha K_{energy} + \beta K_{boundary} + \gamma K_{sparse} + \delta K_{center}. \tag{7}$$

K terms represent the kernel losses that force the kernel extracted from G to be meaningful. K_{energy} make the kernel conserve the energy of the input data; $K_{boundary}$ and K_{sparse} make the kernel not be an over-smoothing kernel; and K_{center} centers the kernel. α, β, γ, and δ are the regularization parameters of K terms, respectively. Because the total variation guiding scheme requires only simple calculations on the input image and no additional network, the proposed method efficiently improves the kernel estimation performance with less additional cost and memory than KernelGAN and E-KernelGAN.

4. Experimental Results

We evaluated our method using three datasets: DIV2KRK, Flickr2KRK and DIV2KSK. The DIV2KRK dataset consists of 100 validation images from DIV2K [26] degraded with random kernels that were generated in [20] to follow an anisotropic Gaussian random distribution and applied by multiplicative noise. Similarly, Flickr2KRK was generated using the first 100 images in Flickr2K [27] by applying the same kernel generation process. In both datasets, we shuffled 100 kernels and used them to degrade and downsample the ground truth (GT) images for scale factors of ×2 and ×4. However, these datasets lack sufficiently sizable and anisotropic kernels, and have meaningless kernels with several isolated peaks. To evaluate the performance of the kernel estimation on sizable and anisotropic kernels, we generated a new dataset named the DIV2KSK (DIV2K Synthetic Kernel). We randomly selected 15 validation images from DIV2K [26], and degraded and downsampled them using 16 synthetic kernels for scale factors of ×2 and ×4, respectively, to produce total 240 input images.

We implemented our algorithm using the Python PyTorch library and trained it using an NVIDIA GeForce RTX 3090 GPU. For training, we set the initial learning rate to 2×10^{-4} and trained the network for 3000 iterations using the ADAM optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The parameters c and s were set to 0.6 and 2, respectively.

4.1. Kernel Estimation Results

We evaluated our method by comparing it with the conventional kernel estimation algorithms; KernelGAN [20], FKP-KernelGAN [21], E-KernelGAN [22] and E-KernelGAN-DIP. The E-KernelGAN-DIP utilizes a deep image prior (DIP) [28] network to estimate more reasonable kernels. To quantitatively evaluate the estimated kernels, we used two metrics: kernel error(KE) and kernel similarity(KS), as follows:

$$\begin{aligned} KE &= \|B^{GT} - \hat{B}\|_2^2, \\ KS &= \frac{B^{GT} \cdot \hat{B}}{\|B^{GT}\|_2 \|\hat{B}\|_2}. \end{aligned} \tag{8}$$

KE is the sum of the difference squares between the GT kernel B^{GT} and the estimated kernel $\hat{B}$. KE represents the errors of the values of $\hat{B}$ to those of B^{GT}. However, it tends to be low when $\hat{B}$ is a round shape and large enough. To address this limitation, we introduce a metric KS similar to that proposed in [29] to evaluate the shape similarity between B^{GT} and $\hat{B}$. To ensure a fair comparison, all kernels, including the ground truth kernels, are moved for their center of mass to be centered because we do not consider image shift. In addition, to analyze the relationship between the kernel estimation performance of the algorithms and the GT kernel size, we introduce the kernel size r as follows:

$$\begin{aligned} M^T &= \begin{cases} 1, & \text{if } B_i > T, \\ 0, & \text{otherwise,} \end{cases} \text{ where } i \in B, T = \frac{\max(B)}{30}, \\ r &= \frac{\sum_i M_i^T}{N_1 N_2}. \end{aligned} \tag{9}$$

M^T is a binary mask where the elements of B greater than the threshold T are marked, i is the location of the kernel space, and (N_1, N_2) is the kernel space size, that is, $(17, 17)$ in the case of a scale factor ×2. The region marked by M^T captures most of the kernel energy of a given kernel (at least 95 percent of the total kernel energy).

First, a qualitative comparison of kernel estimation results on the DIV2KRK and Flickr2KRK datasets for scale factor of ×2 is shown in Figures 5 and 6, respectively. KernelGAN often fails to estimate the direction and overall shape of a kernel, including its length and thickness. FKP-KernelGAN(FKP) attempted to estimate the kernel as closely to the GT kernel as possible, but it had clear limitations, as it could not present kernels on which it had not previously been trained. In the case of E-KernelGAN and E-KernelGAN-DIP, although they could stably estimate the kernel direction, the shape of the estimated kernels tended to be relatively small, round-shaped, and short compared to the GT kernels. Our proposed method, TVG-KernelGAN, estimates kernels that approximate the GT kernels, regarding both the kernel direction and overall shape. However, for small and sharp kernels such as the kernel in the 4th row of Figure 6f, TVG-KernelGAN tended to estimate kernels thicker than the GT kernel. Next, a qualitative comparison of the kernel estimation results on the DIV2KSK dataset for the scale factor of ×2 is shown in Figure 7. The kernel estimation tendencies were similar to those of the two previous two datasets. KernelGAN was unstable and inaccurate, FKP had obvious limitations, and the results of E-KernelGAN and E-KernelGAN-DIP still had insufficient length. By contrast, TVG-KernelGAN outperformed the other conventional methods, as the GT kernels were large and anisotropic.

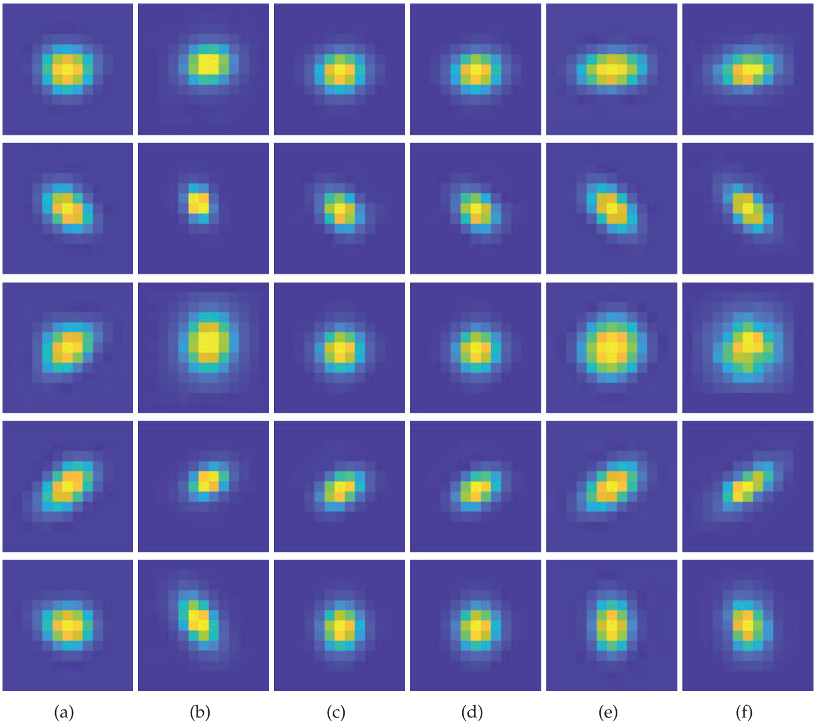


Figure 5. Qualitative results of the kernel estimation for scale factor of ×2 for DIV2KRK dataset. (**a**) KernelGAN [20], (**b**) FKP [21], (**c**) E-KernelGAN [22], (**d**) E-KernelGAN-DIP [22], (**e**) TVG-KernelGAN, (**f**) GT.

A quantitative comparison of the kernel estimation results for the entire dataset and for the scale factors of ×2 and ×4 is shown in Table 1. A lower value of *KE* and a higher value of *KS* indicate better performance. The numbers in red indicate the first-highest scores. For the DIV2KRK dataset, E-KernelGAN-DIP achieved the highest scores for both *KE* and *KS*, followed by E-KernelGAN, which achieved a score almost identical to that of E-KernelGAN-DIP. TVG-KernelGAN achieved the third-highest score. For the Flickr2KRK dataset, TVG-KernelGAN achieved the highest score for *KE* and the second-highest score for *KS*, whereas E-KernelGAN-DIP achieved the second-highest score for *KE* and the highest score for *KS*. E-KernelGAN achieved the third-highest score for both *KE* and *KS*. The E-KernelGAN, E-KernelGAN-DIP, and TVG-KernelGAN scores differ little for the two datasets. However, for the DIV2KSK dataset, TVG-KernelGAN achieved a significantly higher score than the other conventional methods. This quantitative comparison was consistent with the qualitative comparison presented above. In addition, Figure 8a,b, respectively, show the mean of *KE* and *KS* of all three dataset samples according to kernel size r, and Figure 8c shows examples of kernels of various sizes and the corresponding r values. TVG-KernelGAN achieved the highest scores for both *KE* and *KS* except for the smallest kernel size. These results suggest that the proposed method is more accurate and stable than conventional methods for estimating sizable and anisotropic kernels.

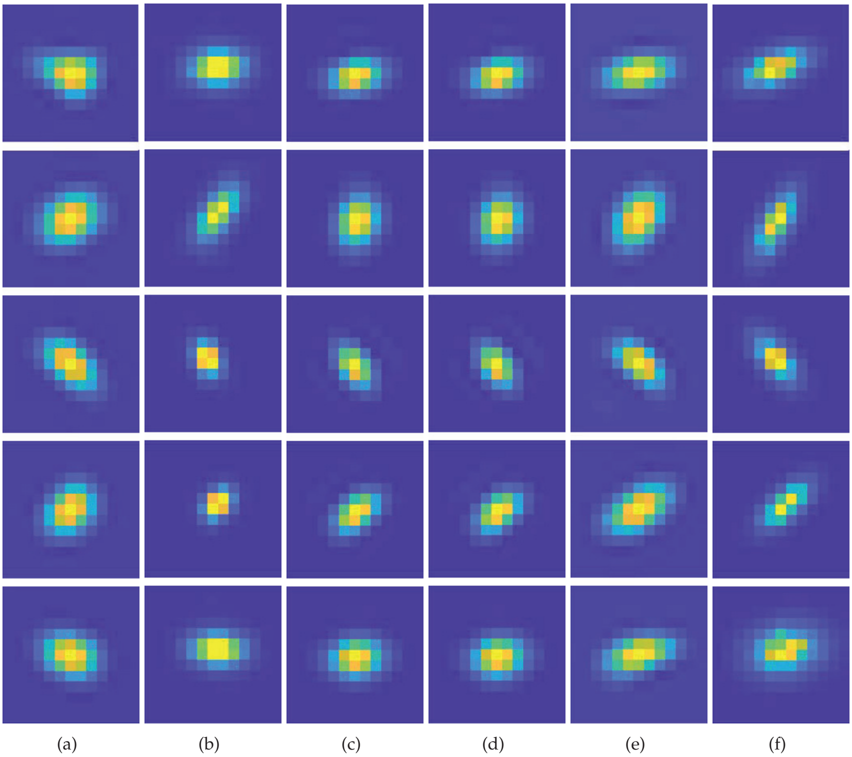


Figure 6. Qualitative results of the kernel estimation for scale factor of ×2 for Flickr2KRK dataset. (**a**) KernelGAN [20], (**b**) FKP [21], (**c**) E-KernelGAN [22], (**d**) E-KernelGAN-DIP [22], (**e**) TVG-KernelGAN, (**f**) GT.

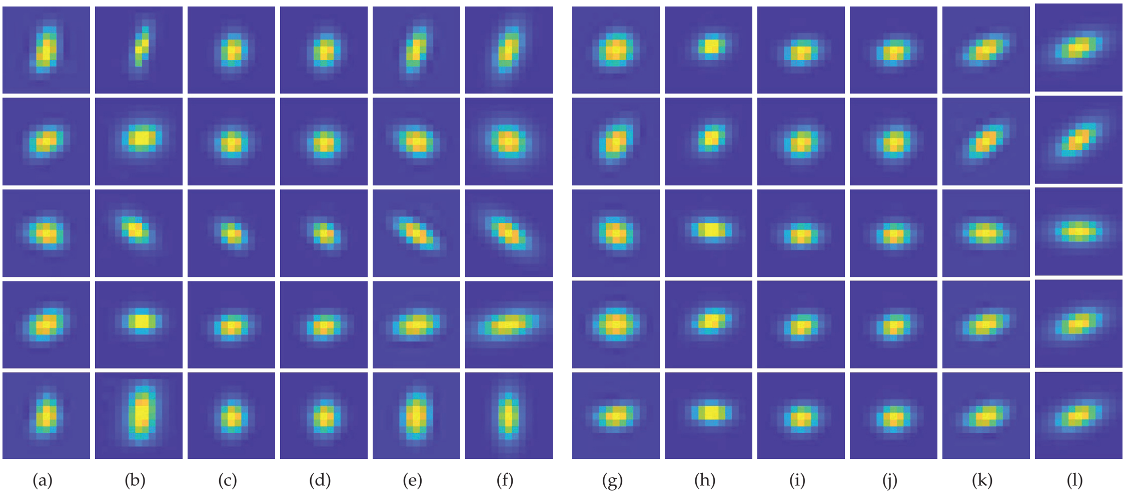
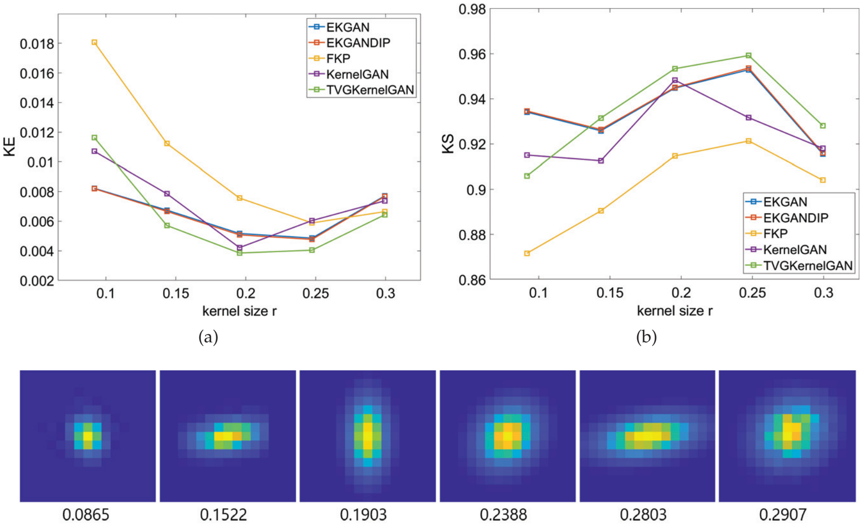


Figure 7. Qualitative results of the kernel estimation for scale factor of ×2 for DIV2KSK dataset. (**a**,**g**) KernelGAN [20], (**b**,**h**) FKP [21], (**c**,**i**) E-KernelGAN [22], (**d**,**j**) E-KernelGAN-DIP [22], (**e**,**k**) TVG-KernelGAN, (**f**,**l**) GT.



(a) (b)

(c)

Figure 8. *KE* and *KS* curves according to the kernel sizes *r*. (**a**) KE, (**b**) KS, (**c**) the kernel examples and the corresponding kernel sizes *r*.

Table 1. Comparison of the kernel estimation results in terms of quantitative score, kernel error (KE) and kernel similarity (KS).

			KernelGAN	**FKP**	**E-KernelGAN**	**E-KernelGAN-DIP**	**TVG-KernelGAN**
DIV2KRK	×2	*KE*	0.0067	0.0072	0.0043	0.0043	0.0046
		KS	0.9294	0.9239	0.9574	0.9579	0.9543
	×4	*KE*	0.00088	0.00080	0.00062	0.00062	0.00070
		KS	0.9537	0.9537	0.9698	0.9699	0.9680
Flickr2KRK	×2	*KE*	0.0087	0.0106	0.0081	0.0080	0.0077
		KS	0.8989	0.8833	0.9094	0.9100	0.9097
	×4	*KE*	0.00111	0.00093	0.00090	0.00089	0.00089
		KS	0.9391	0.9392	0.9550	0.9550	0.9552
DIV2KSK	×2	*KE*	0.0051	0.0072	0.0058	0.0057	0.0043
		KS	0.9446	0.9138	0.9426	0.9431	0.9547
	×4	*KE*	0.00088	0.00100	0.00083	0.00083	0.00074
		KS	0.9478	0.9419	0.9577	0.9579	0.9593

4.2. Non-Blind Super-Resolution Results

In this subsection, we conducted experiments on two branches of SISR, the classical approach and the deep-learning-based approach, to show that the SISR results are improved by using the kernels estimated by the proposed method, particularly for sizable and anisotropic kernels. First, we employed ZSSR [13] as the deep-learning-based approach. Briefly explained, it downscales the input image with a given blurring kernel and then upscales it using a deep-learning upscaling network. It imitates the inverse of the downscaling process for the upscaling network to predict the output SR image after the training session. In this process, the more accurate the blurring kernel is for the downscaling, the higher the SR performance. Next, we employed Equation (2) as a classical approach, by optimizing it using the conjugate gradient descent method. We set $\lambda = 1 \times 10^{-6}$, $p = 0.8$ and ∇ in the regularization term as a forward difference derivative operator.

Qualitative comparisons of the SR results obtained using the two SR methods for the scale factor of ×2 are shown in Figures 9 and 10, and that for the scale factor of ×4 are shown in Figures 11 and 12. We observed that wrongly estimated kernels result in artifacts in the SR results. First, when the estimated kernels were small, the SR results exhibited both the ringing and blurring artifacts as shown in Figures 9d and 10d. Second, when the estimated kernels had the incorrect anisotropic direction or round shapes, the SR results exhibited ringing artifacts as shown in Figures 10c,e,f, 11c,e,f and 12c. Lastly, when the estimated kernels had the correct anisotropic direction but insufficient length, the SR results again exhibited ringing artifacts as shown in Figures 9e,f and 10d, or the slightly blurry artifacts as shown in Figure 12e,f. In contrast, the SR results using the kernels estimated by the proposed method showed resolution enhancement with much less or no ringing artifacts as shown in Figures 9g, 10g, 11g and 12g. For a quantitative comparison, we also measured PSNR and SSIM between the GT images and the SR results images for the entire dataset and the scale factor of ×2 as shown in Table 2. E-KernelGAN and E-KernelGAN-DIP achieved superior or similar scores with TVG-KernelGAN for the first two datasets. However, TVG-KernelGAN achieved superior scores for the DIV2KSK dataset with more sizable and anisotropic kernels. These results are consistent with that in Section 4.1, showing that the performance of the SR algorithms using the sizable and anisotropic kernels estimated by the proposed method has been improved.

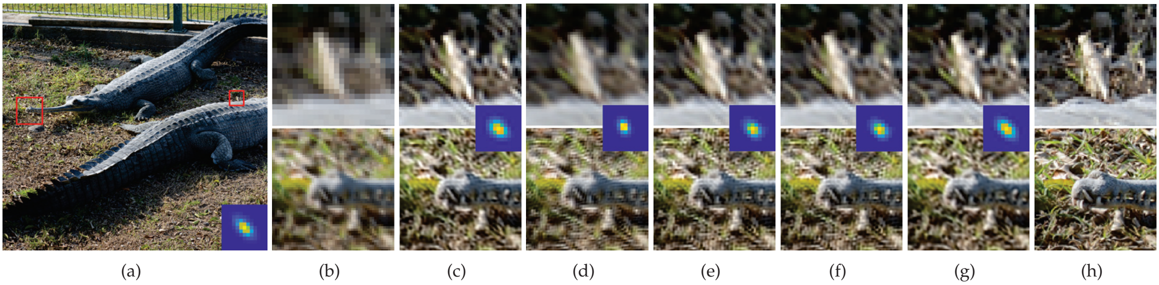


Figure 9. SR results of the 95th image of DIV2KRK dataset using the estimated kernels for scale factor of ×2. The first row is the ZSSR results and the second row is the results of solving Equation (2). From (**c**) to (**g**), the estimated kernels are shown in the middle of each column. (**a**) GT image and kernel, (**b**) LR, (**c**) KernelGAN [20], (**d**) FKP [21], (**e**) E-KernelGAN [22], (**f**) E-KernelGAN-DIP [22], (**g**) TVG-KernelGAN, (**h**) GT.

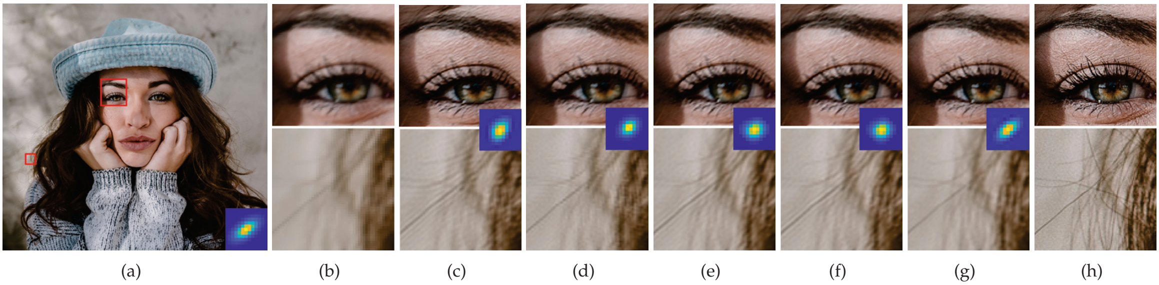


Figure 10. SR results of the 8th image degraded by the 12th kernel of DIV2KSK dataset using the estimated kernels for scale factor of ×2. The first row is the ZSSR results and the second row is the results of solving Equation (2). From (**c**) to (**g**), the estimated kernels are shown in the middle of each column. (**a**) GT image and kernel, (**b**) LR, (**c**) KernelGAN [20], (**d**) FKP [21], (**e**) E-KernelGAN [22], (**f**) E-KernelGAN-DIP [22], (**g**) TVG-KernelGAN, (**h**) GT.

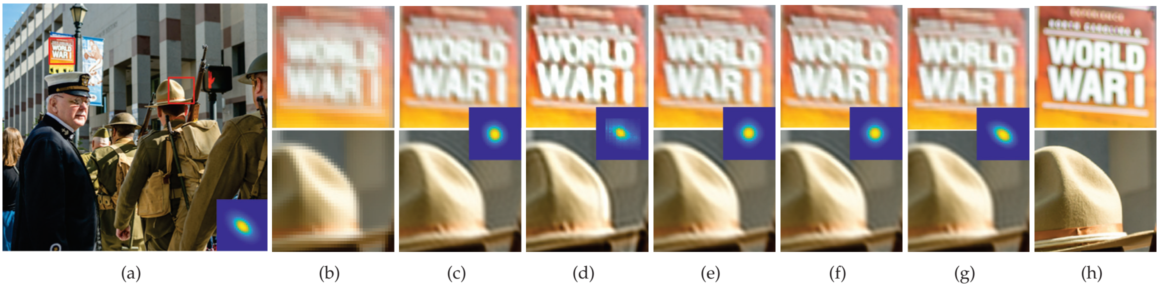


Figure 11. SR results of the 34th image of Flickr2KRK dataset using the estimated kernels for scale factor of ×4. The first row is the ZSSR results and the second row is the results of solving Equation (2). From (**c**) to (**g**), the estimated kernels are shown in the middle of each column. (**a**) GT image and kernel, (**b**) LR, (**c**) KernelGAN [20], (**d**) FKP [21], (**e**) E-KernelGAN [22], (**f**) E-KernelGAN-DIP [22], (**g**) TVG-KernelGAN, (**h**) GT.

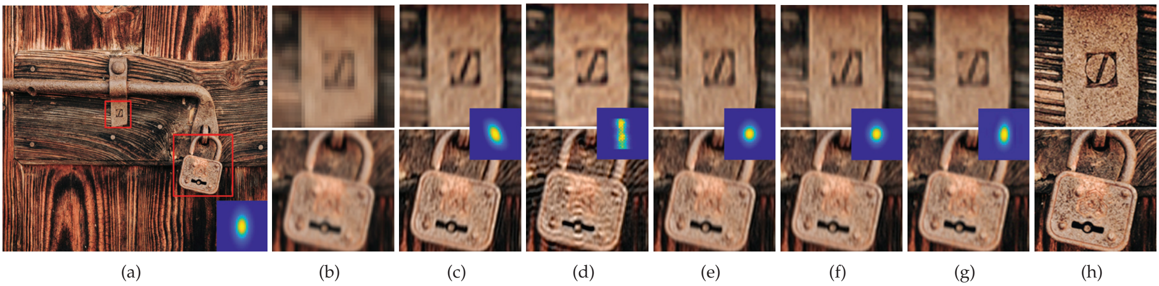


Figure 12. SR results of the 14th image degraded by the 1st kernel of DIV2KSK dataset using the estimated kernels for scale factor of ×4. The first row is the ZSSR results and the second row is the results of solving Equation (2). From (**c**) to (**g**), the estimated kernels are shown in the middle of each column. (**a**) GT image and kernel, (**b**) LR, (**c**) KernelGAN [20], (**d**) FKP [21], (**e**) E-KernelGAN [22], (**f**) E-KernelGAN-DIP [22], (**g**) TVG-KernelGAN, (**h**) GT.

Table 2. Comparison of PSNR and SSIM scores of the SR results using estimated kernels.

			Bicubic	**KernelGAN**	**FKP**	**E-KernelGAN**	**E-KernelGAN-DIP**	**TVG-KernelGAN**	**GT**
DIV2KRK	ZSSR	PSNR	28.6953	28.7329	28.3635	29.3803	29.3544	29.0642	29.8799
		SSIM	0.8035	0.8360	0.8413	0.8472	0.8470	0.8416	0.8656
	Equation (2)	PSNR	28.6953	30.0237	28.9431	30.3637	30.3741	30.3425	31.5232
		SSIM	0.8035	0.8516	0.8329	0.8573	0.8576	0.8562	0.8801
Flickr2KRK	ZSSR	PSNR	28.0653	28.4859	27.5576	28.7836	28.7809	28.5700	29.4258
		SSIM	0.7897	0.8230	0.8281	0.8297	0.8296	0.8281	0.8500
	Equation (2)	PSNR	28.0653	29.2672	28.5526	29.3507	29.3542	29.2909	29.0367
		SSIM	0.7897	0.8385	0.8222	0.8404	0.8406	0.8406	0.8341
DIV2KSK	ZSSR	PSNR	24.5548	25.3626	24.7414	25.4244	25.4294	25.4313	26.2298
		SSIM	0.6874	0.7507	0.7514	0.7499	0.7496	0.7529	0.7921
	Equation (2)	PSNR	24.5548	25.9113	25.1470	25.7892	25.7525	25.9618	26.8826
		SSIM	0.6874	0.7608	0.7300	0.7566	0.7563	0.7629	0.7975

4.3. Memory and Cost Efficiency

We evaluated the cost and memory efficiency of the KernelGAN series, including KernelGAN, E-KernelGAN, E-KernelGAN-DIP, and TVG-KernelGAN, by measuring the parameter numbers and run-time for 3000 iterations at a scale factor of ×2. The results of these measurements are presented in Table 3. First, owing to the use of a large *D* network, E-KernelGAN had parameters that were 2.5 times more than that of KernelGAN, and required several times more run-time for kernel estimation than KernelGAN. Furthermore, as E-KernelGAN-DIP utilizes the DIP network, it has significantly more parameters and requires a much longer time, as shown in Table 3. However, because we did not construct any additional networks, TVG-KernelGAN has the same parameter numbers as KernelGAN and takes same time as KernelGAN, making it much more time-efficient than E-KernelGAN and E-KernelGAN-DIP. These results suggest that the TVG-KernelGAN can efficiently leverage self-similarity in the input image with a simple modification.

Table 3. Run-time for scale factor of ×2 and the network parameters of KernelGAN series.

	KernelGAN	**E-KernelGAN**	**E-KernelGAN-DIP**	**TVG-KernelGAN**
Network parameters	181 k	464 k	2824 k	181 k
Run-time	57 s	356 s	930 s	57 s

4.4. Limitation

As shown in Sections 4.1 and 4.2, TVG-KernelGAN performed better in estimating sizable and anisotropic kernels. However, it achieved lower scores of KE and KS for the smallest kernel size *r* as shown in Figure 8, achieving even lower scores than those of KernelGAN. In a comparison of PSNR and SSIM scores in Table 2, TVG-KernelGAN showed lower scores than those of the E-KernelGAN series, particularly on the DIV2KRK dataset because the DIV2KRK dataset has many small-size kernels compared to the other two datasets. To our knowledge, TVG-KernelGAN fails to estimate the small-size kernels because we emphasized the edge region of the input image rather than the edge itself to estimate sizable and anisotropic kernels. The input image is less blurry when the degradation blurring kernel is small. Then, the small kernel from *G* can easily minimize the GAN loss by utilizing the relatively sharp edge of the original input image. On the contrary, weighting the edge region makes the original edge thicker, preventing *G* from estimating the small kernel. Therefore, we expect that an adaptive algorithm that utilizes the edge weighting scheme according to the degree of smoothness will help solve this problem.

5. Conclusions

In this study, we proposed a kernel estimation method for image super-resolution using GAN guided by a total variation map. We simply weighted the input image using its total variation, which includes four directions, to emphasize the edge region, which has prevalent structural information for the network efficiently to maximize the self-similarity of the given input image. The experimental results, including the qualitative and quantitative evaluations, demonstrate that the proposed method estimates the SR kernels more accurately and stably than conventional methods, particularly for sizable and anisotropic kernels. The super-resolution results further show that the proposed method is superior to the compared methods. In addition, the network parameter numbers and run-time measurements demonstrate the efficiency of the proposed method, which simply modifies the input data.

Author Contributions: Conceptualization, J.P. and H.K.; methodology, J.P.; software, J.P.; validation, J.P., H.K. and M.G.K.; supervision, J.P. and M.G.K.; funding acquisition, M.G.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A2C200289711).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Park, S.C.; Park, M.K.; Kang, M.G. Super-resolution image reconstruction: A technical overview. *IEEE Signal Process. Mag.* **2003**, *20*, 21–36. [CrossRef]
2. Farsiu, S.; Robinson, M.D.; Elad, M.; Milanfar, P. Fast and robust multiframe super resolution. *IEEE Trans. Image Process.* **2004**, *13*, 1327–1344. [CrossRef] [PubMed]
3. Hardie, R.C.; Barnard, K.J.; Bognar, J.G.; Armstrong, E.E.; Watson, E.A. High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. *Opt. Eng.* **1998**, *37*, 247–260.
4. Yuan, Q.; Zhang, L.; Shen, H. Multiframe super-resolution employing a spatially weighted total variation model. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *22*, 379–392. [CrossRef]
5. Köhler, T.; Huang, X.; Schebesch, F.; Aichert, A.; Maier, A.; Hornegger, J. Robust multiframe super-resolution employing iteratively re-weighted minimization. *IEEE Trans. Comput. Imaging* **2016**, *2*, 42–58. [CrossRef]
6. Glasner, D.; Bagon, S.; Irani, M. Super-resolution from a single image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 349–356.
7. Michaeli, T.; Irani, M. Nonparametric Blind Super-resolution. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Nice, France, 13 December 2013.
8. Freeman, W.T.; Jones, T.R.; Pasztor, E.C. Example-based super-resolution. *IEEE Comput. Graph. Appl.* **2002**, *22*, 56–65. [CrossRef]
9. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
10. 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, NA, USA, 27–30 June 2016; pp. 1646–1654.
11. 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.
12. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep back-projection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1664–1673.
13. Shocher, A.; Cohen, N.; Irani, M. "zero-shot" super-resolution using deep internal learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3118–3126.
14. 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.
15. Ma, C.; Rao, Y.; Cheng, Y.; Chen, C.; Lu, J.; Zhou, J. Structure-preserving super resolution with gradient guidance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7769–7778.
16. 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, 14–19 June 2020; pp. 466–467.
17. Lugmayr, A.; Danelljan, M.; Timofte, R. Unsupervised learning for real-world super-resolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3408–3416.
18. Lugmayr, A.; Danelljan, M.; Timofte, R.; Fritsche, M.; Gu, S.; Purohit, K.; Kandula, P.; Suin, M.; Rajagoapalan, A.; Joon, N.H.; et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3575–3583.
19. Lugmayr, A.; Danelljan, M.; Timofte, R. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Nashville, TN, USA, 19–25 June 2020; pp. 494–495.
20. Bell-Kligler, S.; Shocher, A.; Irani, M. Blind Super-Resolution Kernel Estimation using an Internal-GAN. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
21. Liang, J.; Zhang, K.; Gu, S.; Van Gool, L.; Timofte, R. Flow-based kernel prior with application to blind super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10601–10610.
22. Kim, Y.; Ha, J.; Cho, Y.; Kim, J. Unsupervised Blur Kernel Estimation and Correction for Blind Super-Resolution. *IEEE Access* **2022**, *10*, 45179–45189. [CrossRef]
23. Yamac, M.; Ataman, B.; Nawaz, A. KernelNet: A Blind Super-Resolution Kernel Estimation Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nashville, TN, USA, 19–25 June 2021; pp. 453–462.

24. Cho, S.; Lee, S. Fast Motion Deblurring. In Proceedings of the ACM SIGGRAPH Asia 2009 Papers, SIGGRAPH Asia '09, Yokohama, Japan, 16–19 December 2009; Association for Computing Machinery: New York, NY, USA, 2009. [CrossRef]
25. Shan, Q.; Jia, J.; Agarwala, A. High-Quality Motion Deblurring from a Single Image. *ACM Trans. Graph.* **2008**, *27*, 1–10. [CrossRef]
26. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017.
27. 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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017.
28. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep image prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9446–9454.
29. Hu, Z.; Yang, M.H. Good Regions to Deblur. In Proceedings of the Computer Vision—ECCV 2012, Florence, Italy, 7–13 October 2012; Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 59–72.

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

Communication

Identifying the Edges of the Optic Cup and the Optic Disc in Glaucoma Patients by Segmentation

Srikanth Tadisetty [1], Ranjith Chodavarapu [1], Ruoming Jin [1], Robert J. Clements [2] and Minzhong Yu [3,*]

[1] Department of Computer Science, Kent State University, Kent, OH 44242, USA; stadiset@kent.edu (S.T.); rchodava@kent.edu (R.C.)
[2] Department of Biological Sciences, Kent State University, Kent, OH 44242, USA; rclement@kent.edu
[3] Department of Ophthalmology, University Hospitals, Case Western Reserve University, Cleveland, OH 44106, USA
* Correspondence: minzhong.yu@uhhospitals.org

Abstract: With recent advancements in artificial intelligence, fundus diseases can be classified automatically for early diagnosis, and this is an interest of many researchers. The study aims to detect the edges of the optic cup and the optic disc of fundus images taken from glaucoma patients, which has further applications in the analysis of the cup-to-disc ratio (CDR). We apply a modified U-Net model architecture on various fundus datasets and use segmentation metrics to evaluate the model. We apply edge detection and dilation to post-process the segmentation and better visualize the optic cup and optic disc. Our model results are based on ORIGA, RIM-ONE v3, REFUGE, and Drishti-GS datasets. Our results show that our methodology obtains promising segmentation efficiency for CDR analysis.

Keywords: segmentation; edge; detection; eye; diseases; glaucoma

Citation: Tadisetty, S.; Chodavarapu, R.; Jin, R.; Clements, R.J.; Yu, M. Identifying the Edges of the Optic Cup and the Optic Disc in Glaucoma Patients by Segmentation. *Sensors* **2023**, *23*, 4668. https://doi.org/10.3390/s23104668

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 13 March 2023
Revised: 8 May 2023
Accepted: 10 May 2023
Published: 11 May 2023

1. Introduction

Fundus images are routinely used to detect eye diseases. Ophthalmologists used to analyze these images via a non-automated process, and it is a heavy burden for any ophthalmologist to read and explain the fundus images during the diagnosis of the ocular diseases [1]. Glaucoma is one of the major ocular diseases that causes visual impairment [2]. According to the World Health Organization (WHO), it has affected millions of people globally, and the early detection of glaucoma can prevent vision loss. The optic nerve transfers signals from the retina to the brain, whereby ganglion cell axons converge at the optic disc and exit the eye to form the optic nerve. The optic disc has a cup-shaped structure at the center, called the optic cup, which has a different color than the optic disc. In individuals with glaucoma, the size of the optic cup increases due to the death of the ganglion cells caused by the increase in intraocular pressure (IOP) and/or the loss of blood flow to the optic nerve. Therefore, the cup-to-disc ratio (CDR) is a main index for the early diagnosis of glaucoma and for the quantitative evaluation of the severity of glaucoma. The normal CDR is less than 0.5. A CDR less than 0.4 without an abnormally small optic disc size indicates a normal optic disc. In this stage, glaucoma must be diagnosed by IOP or other methods. If the CDR is between 0.5 and 0.8, it is considered the moderate stage of glaucoma. If the CDR is higher than 0.8, it is considered the severe stage of glaucoma [3]. With recent advancements in artificial intelligence, fundus images with different diseases can be classified automatically for the early diagnosis of diseases. The most widely used method in image classification networks is the application of convolutional neural networks. Many previous studies have used various pre-trained network architectures for the classification of images and various other methods to obtain the edges of the optic cup and the optic disc in fundus images. In the current study, several datasets of glaucoma fundus images were segmented and compared using our proposed deep learning methodology. U-Net is particularly effective for biomedical image segmentation tasks, such as cell and tissue segmentation [4]. U-Net outperforms other CNN architectures, such as VGG and ResNet, in these applications affirming its potential utility for the current task [4].

In this paper, we propose a new method to visualize the contours of the optic cup and disk. We implemented a modified U-Net for segmenting the optic cup and optic disc of the glaucoma images, later applying edge detection and dilation using the Canny edge filter. Our model is evaluated on four publicly available datasets namely ORIGA, RIM-ONE v3, REFUGE, and Drishti-GS. Our approach achieves a good performance measured using popular image segmentation metrics (IOU and Dice) in detecting early-stage CDR.

2. Related Work

Several studies have aimed to segment fundus images. Among them, Cheng et al. are the first to utilize a clustering-based approach for the segmentation of both the optic disc and optic cup [5]. Sarkar et al. proposed the threshold-based approach for the segmentation of both the optic disc and optic cup on the RIM-ONE dataset [6]. Sun et al. used a deep object detection network for the joint localization and segmentation of the optic cup and disc on the ORIGA dataset [7]. Thakur et al. used a level-set based approach to adaptively regularize Kernel-based intuitionistic Fuzzy C means (LARKIFCM) for optic cup and disc segmentation on RIM-ONE and Drishti-GS datasets [8]. Sevastopolsky et al. used a modified U-Net for disc and cup segmentation on RIM-ONE-V3 and DRISHTI-GS datasets [9]. Kim et al. used an FCN (fully connected network) on the RIGA dataset [10]. Yu et al. uses Modified U-net from ResNet-34 for segmentation on Messidor and RIGA datasets [11]. Al-Bande et al. used Fully conventional Dense-Net for disc and cup segmentation [12].

Some recent studies consider adopting the state-of-the-art deep vision architectures. Guo et al. segmented the optic cup and optic disc of glaucoma images using segmentation models, such as DeepCDR, Wavelet, and their proposed modified U-Net++ [13]. Fu et al. segmented the disc and cup in glaucoma using polar transformation and the deep learning architecture named M-net. That network solves the segmentation of the optic disc and the optic cup in a single-stage multi-layer input and is shown to perform better on the ORIGA and SCES datasets compared to other segmentation models, such as U-net, Superpixel, LRR, etc. [14]. Bajwa et al. used G1020, a large publicly available dataset with 1020 fundus images for glaucoma classification. They obtained an accuracy of approximately 80% using the Inception V3 architecture [15]. Anitha et al. classified and segmented the glaucoma images using a trained DenseNet-201 classifier and U-Net segmentation model. They show their models perform better than other deep learning models, such as VGG19, Inception, ResNet, etc., on ORIGA dataset [16]. Juneja et al. segmented the optic disc and cup using a modified version of the U-Net architecture and tested on the DRISHTI-GS dataset [17]. Pascal et al. developed a model that simultaneously learns the segmentation and classification and tested on REFUGE [18]. Jiang et al. used a region-based convolutional neural network for joint optic cup and optic disc segmentation, which was shown to outperform other methods on the ORIGA dataset [19]. Gu et al. proposed a context encoder network, which gathered high-level data and saved them as spatial data for segmentation and was shown to perform better on DRIVE datasets [20]. Liu et al. proposed a multi-layer edge attention network that utilizes the edge information in the encoding stage [21]. Bajwa et al. evaluated the disc localization on the ORIGA dataset, which resulted in a 2.7% relative improvement over the state-of-the-art results on the ORIGA dataset [22]. Xie et al. proposed a novel fully convolutional network called SU-Net, which combines with the Viterbi algorithm to jointly decode the segmentation boundary [23]. Gao et al. developed a Recurrent Fully Convolution Network (RFC-Net) for the automatic joint segmentation of the optic disc and the optic cup, which can capture more high-level information and subtle edge information [24]. Hervella et al. developed a simultaneous classification of glaucoma and segmentation of the optic disc and cup by taking advantage of both pixel-level and image-level labels during network training. Additionally, the segmentation results allowed the extraction of relevant biomarkers such as the cup-to-disc ratio. They have evaluated the model using REFUGE and DRISHTI-GS datasets [25]. Parkhi et al. utilized DeepLabv3 and ensemble models to perform the segmentation of the optic disc and cup [26]. Zhou et al. developed a one-stage network named EfficientNet and Attention-based Residual Depth-Wise Separable Convolution (EARDS) for joint OD and OC segmentation [27]. Wu et al. developed a transformer-based conditional U-Net framework and a new Spectrum-Space Transformer to model the interaction between noise and semantic features. This architectural improvement leads to a new diffusion-based medical image segmentation method called MedSegDiff-V2 [28]. Sun et al. used ResFPN-Net to learn the boundary features and the inner relation between OD and OC for automatic segmentation [29]. Xue et al. used hybrid level set modeling for disc segmentation [30]. Zaaboub et al. proposed a two-stage (OD localization and segmentation) approach to detect the contour of the OD [31]. Liu et al. proposed a novel unsupervised model based on adversarial learning to perform the optic disc and cup segmentation [32]. Xiong et al. proposed a weak label-based Bayesian U-Net exploiting Hough transform-based annotations to segment the optic disc in fundus images. To achieve this, they built a probabilistic graphical model and explored a Bayesian approach with the state-of-the-art U-Net framework [33]. Wang et al. extended the EfficientNet-based U-Net, named EE-U-Net, for OD and OC segmentation [34].

3. Materials and Methods

3.1. Dataset

In this study, we introduce a modified U-Net model to perform edge segmentation and dilation (boundary thickening) using various datasets with different image resolutions: ORIGA (2499 × 2048), RIM-ONE v3 (1300 × 1100), REFUGE (2124 × 2056), and Drishti-GS (2049 × 1751) (Table 1). Datasets consist of images and masks, which are binary images consisting of zero-valued RGB pixels as background and RGB values greater than or equal to [128, 128, 128] at each pixel index *i* for objects of interest, keeping in mind the presence of gray and white labels.

Table 1. Glaucoma segmentation datasets.

Dataset	Description	Reference
Drishti-GS [35,36]	It contains a total of 101 images. "http://cvit.iiit.ac.in/projects/mip/drishti-gs/mip-dataset2/Home.php (accessed on 19 March 2023)"	[11,37–40]
ORIGA	It has a total of 650 retinal images that are available publicly on Kaggle. "https://www.kaggle.com/datasets/arnavjain1/glaucoma-datasets?select=ORIGA (accessed on 19 March 2023)"	[38,40–43]
RIM-ONE-V3 [44]	RIM-ONE is a publicly available dataset of 74 colored fundus images. "http://medimrg.webs.ull.es/research/downloads/ (accessed on 19 March 2023)"	[37,38,40,45]
REFUGE [46]	It comprises 1200 colored retinal images with 400 images each for testing, validation, and training purposes. "https://www.kaggle.com/datasets/arnavjain1/glaucoma-datasets?select=REFUGE (accessed on 19 March 2023)"	[41]

3.2. Architecture

We use U-Net to extract features from the input fundus images and then convert the features into a high-level visual representation, which are processed for edge detection and dilation. The U-Net consists of an encoder and decoder. The encoder creates a compact representation of the input image (low dimension representation) to extract features via the convolution and pooling layers. The image is upsampled using the decoder, which reconstructs an image from the low dimensional representation. It too consists of the convolution block but has deconvolution layers to increase image dimensionality. The skip connections are the connections between the encoder and decoder that pass earlier features to the decoder. This helps the network capture the input an image's low-level and high-level features. The skip connections are achieved by concatenating the encoder's feature maps with the decoder's corresponding feature maps at the same spatial resolution after the deconvolution [3]. After the initial convolution, the number of channels increases to 64. After the transposed convolution, the image is upsized from 28 × 28 × 1024 to 56 × 56 × 512 and concatenated with the contraction path skip connection image. The final layer is a 1 × 1 convolution to decrease the number of channels without affecting the image resolution (Figure 1). We limit the number of kernels to three for each layer convolution and implement a few pre-processing resizings to downsample the image and improve processing time.

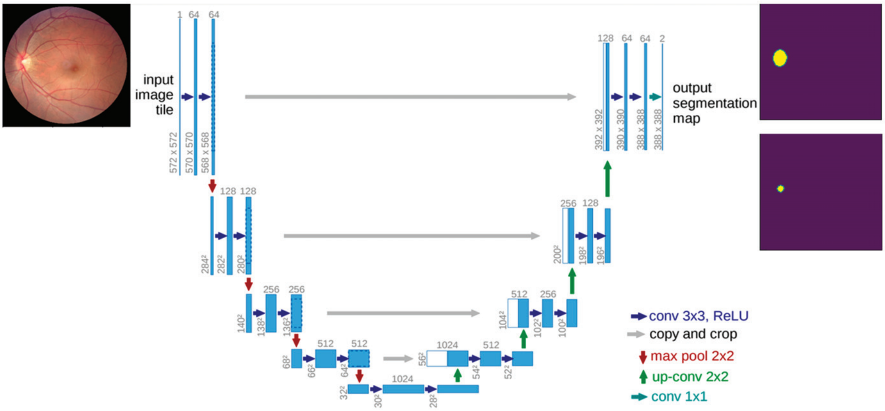


Figure 1. The network architecture of U-Net [3], including the input layer and convolution layer. Due to the size of the figure, the feature dimension is not scaled.

3.3. Evaluation Criteria

Two widely used performance metrics were used for evaluating the segmentation results of the proposed model: (a) Dice Coefficient/F1 Score; (b) Jaccard Score/Intersection over union. The IoU represents the overlapping ratio between the segmentation results and ground truth mask. Both (a) and (b) are positively correlated.

(a) Dice Coefficient: Twice the area of the overlap divided by the total number of the pixels in both images (A and B).

$$DC = \frac{2TP}{2TP + FP + FN} = \frac{2|A \cap B|}{2|A \cap B| + |B \backslash A| + |A \backslash B|} \tag{1}$$

where a true positive is represented by TP, a false positive by FP, and a false negative by FN [27].

(b) Jaccard Score: The area of overlap between the predicted image and the ground truth is divided by the area of union between the predicted image (A) and ground truth image (B).

$$JAC = \frac{TP}{TP + FP + FN} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2}$$

where a true positive is represented by TP, a false positive is represented by FP, and a false negative is represented by FN [27].

3.4. Edge Detection

Canny is an edge detection operator that uses a multistage algorithm. It is composed of five steps: noise reduction, gradient calculation, non-maximum suppression, double threshold, and edge tracking by hysteresis. Noise is removed from the image by applying Gaussian blur via Gaussian kernels. Edges correspond to pixel intensity changes, which are detected by applying filters that highlight intensity changes in different directions (x,y). Non-max suppression is used to thin out the edges by going through all the points in the matrix ((i, j − 1), (i, j + 1), (i + 1,j), and (i − 1,j)) and suppressing (zeroing) non-max pixels. Double thresholding categorizes pixels into strong, weak, and other using a bounding threshold. The hysteresis will then transform weakly categorized pixels into strong ones [47] (Figure 2). Our method then applies a dilation on the resultant image to brighten the Canny generated edge. This entails convolving an image with a kernel that has a defined center. The max overlap pixel overlapped by the kernel is added to the image pixel at the kernel center position, thereby increasing the brightness [47].

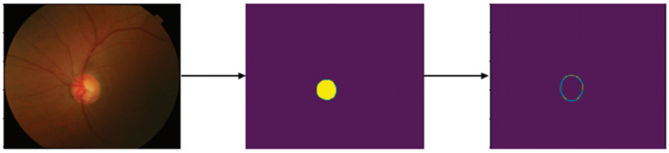


Figure 2. Sample result of Canny without dilation.

4. Experimental Results

In this section, we present the various pre-processing steps and final output results from the different datasets. We carry out the experiments on an Intel California, USA manufactured Intel Xeon Platinum 8268 CPU @ 2.90 GHz running CentOS Stream 8 system with four Nvidia RTX 3090 GPUs having 24 GB of RAM. Each model is run for 300 epochs with a batch size of 4 using an Adam optimizer with a learning rate of 1×10^{-4}.

4.1. Pre-Processing

Datasets of various image dimensions are first resized to 256 × 256 for faster GPU processing. The ORIGA and REFUGE fundus images contained masks that have the cup and disc represented together. Since we are applying segmentation separately without having the cup segmentation hinder the disc or vice versa, the images are separated by changing the pixel values. White pixels are given the gray pixels' values to form the disc images, and vice versa, to generate the cup images (Figure 3). This process was performed with the training, validation, and testing having an 80–10–10% data split, respectively.

Data masks consist of RGB pixel values [0, 0, 0] for the background, and since we have white as cup and grey as disc, the model treats pixels greater than or equal to [128, 128, 128] as object labels (Figure 3).

To avoid overfitting on all datasets, training images were augmented with a random crop generated using the window width and height generated from a normal distribution, Gaussian blur, and random flip. All training, validation, and testing images were then normalized with pixel values between [−1, 1] after first resizing the image to 128 × 128 for model input.

4.2. Segmentation Results

We visualize the loss decrease over the training epochs and the accuracy (Jaccard Score) function curves for each dataset (Figures 4 and 5, respectively). The loss steadily decreases except for the RIM-ONE-V3 dataset, which only consists of 74 images. This is the same result for the accuracy measure over 300 epochs.

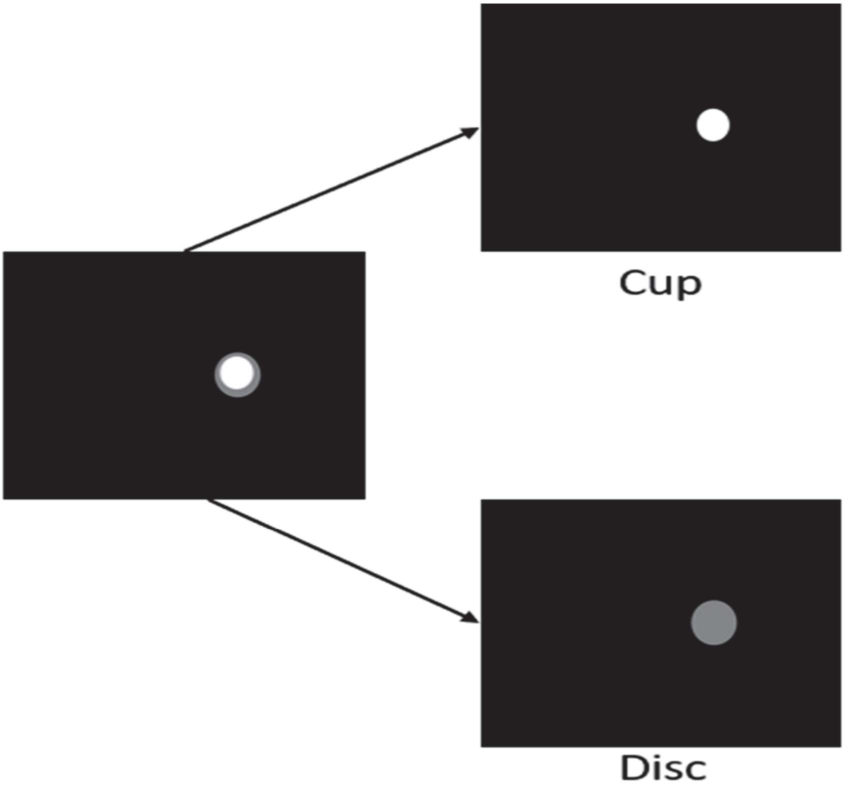


Figure 3. Pre-processing of the mask of the combined ORIGA and REFUGE datasets into separate cup and disc masks.

Table 2 presents the Dice and IoU scores for each dataset using our model consisting of training parameters. Our model has 5,680,865 trainable parameters with 0 untrainable parameters and no frozen/dropped network nodes. On the Drishti-GS dataset, our approach achieves 0.058 and 0.117 for the best validation loss for the disc and cup, respectively. On the RIM-One-V3, it achieves 0.093 and 0.249 for the disc and cup, respectively. ORIGA achieves 0.037 and 0.137 for the disc and cup, respectively. Lastly, on the REFUGE dataset, validation loss achieves a minimal of 0.035 and 0.102 for disc and cup, respectively.

With the Drishti-GS dataset, our approach achieves a 0.943 Dice and 0.893 IoU for OD segmentation. For OC segmentation, it achieves 0.889 Dice and 0.801 IoU. Using the RIM-One-V3 dataset, it obtains 0.910, 0.838 for Dice and IoU, respectively for OD segmentation, and it obtains 0.649, 0.77 for OC segmentation. With the ORIGA dataset, it achieves 0.962 and 0.928 for Dice and IoU, respectively for OD segmentation, and it obtains 0.871, 0.773 for OC segmentation. Lastly, with the REFUGE dataset, it acquires the scores of 0.965 and 0.933 (Dice and IoU respectively) for OD segmentation. This is followed by 0.902 and 0.824 for OC segmentation.

For visualizing the segmentation results, we randomly select images for all testing outputs from Drishti-GS, RIM-One-V3, ORIGA, and REFUGE. Refer to Figures 6 and 7. Figure 6 shows the raw segmentation results without Canny and dilation applied. The first column is the prediction, followed by the ground truth, and the original image is to the right. The optic cup and disc segmentations have separate visualizations. Figure 7 shows the same results from Figure 6 with Canny and dilation applied to the resultant raw segmentation from our model output.

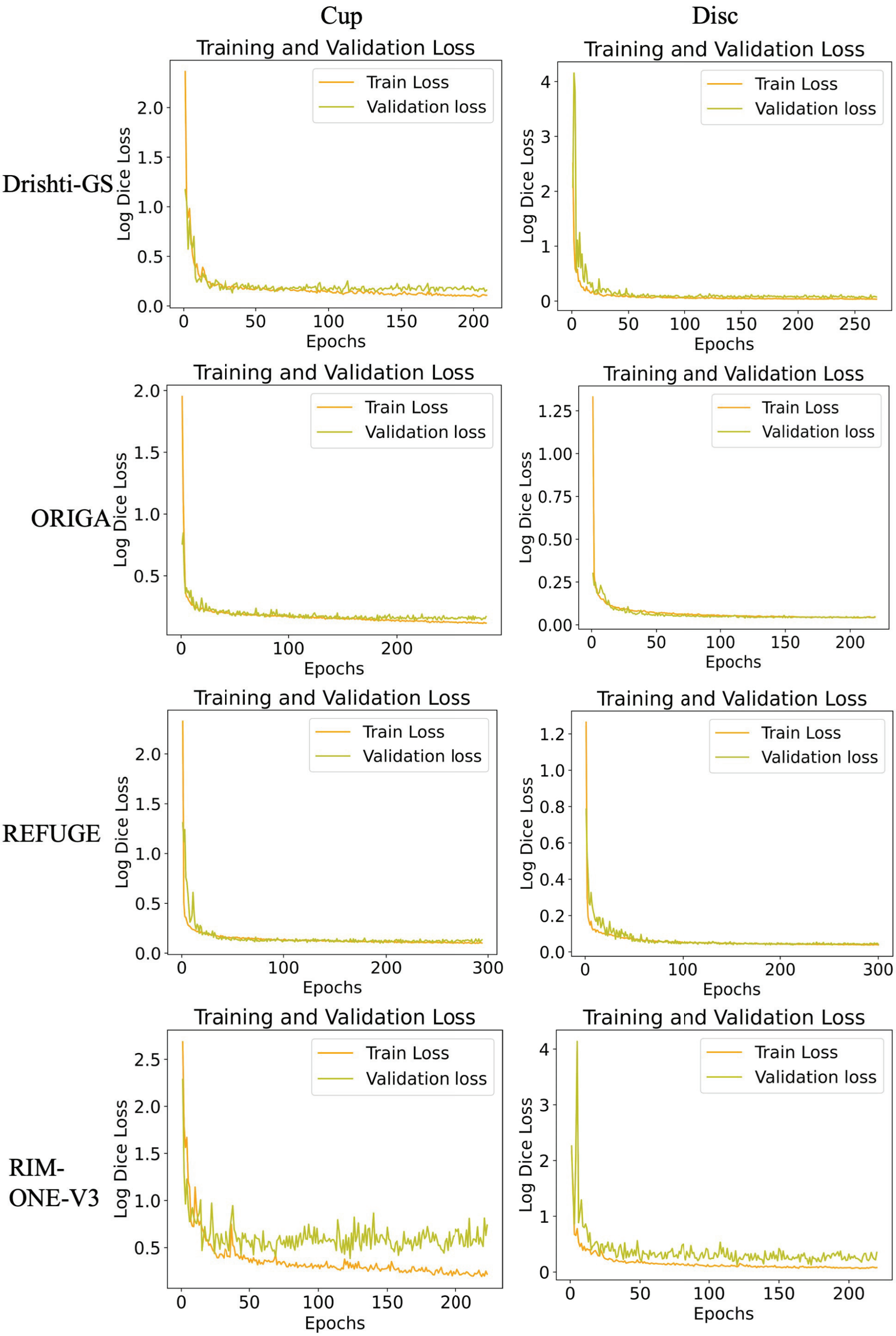


Figure 4. Loss function curves on various datasets (epoch vs. log Dice loss). Orange is the training loss and yellow is the validation loss.

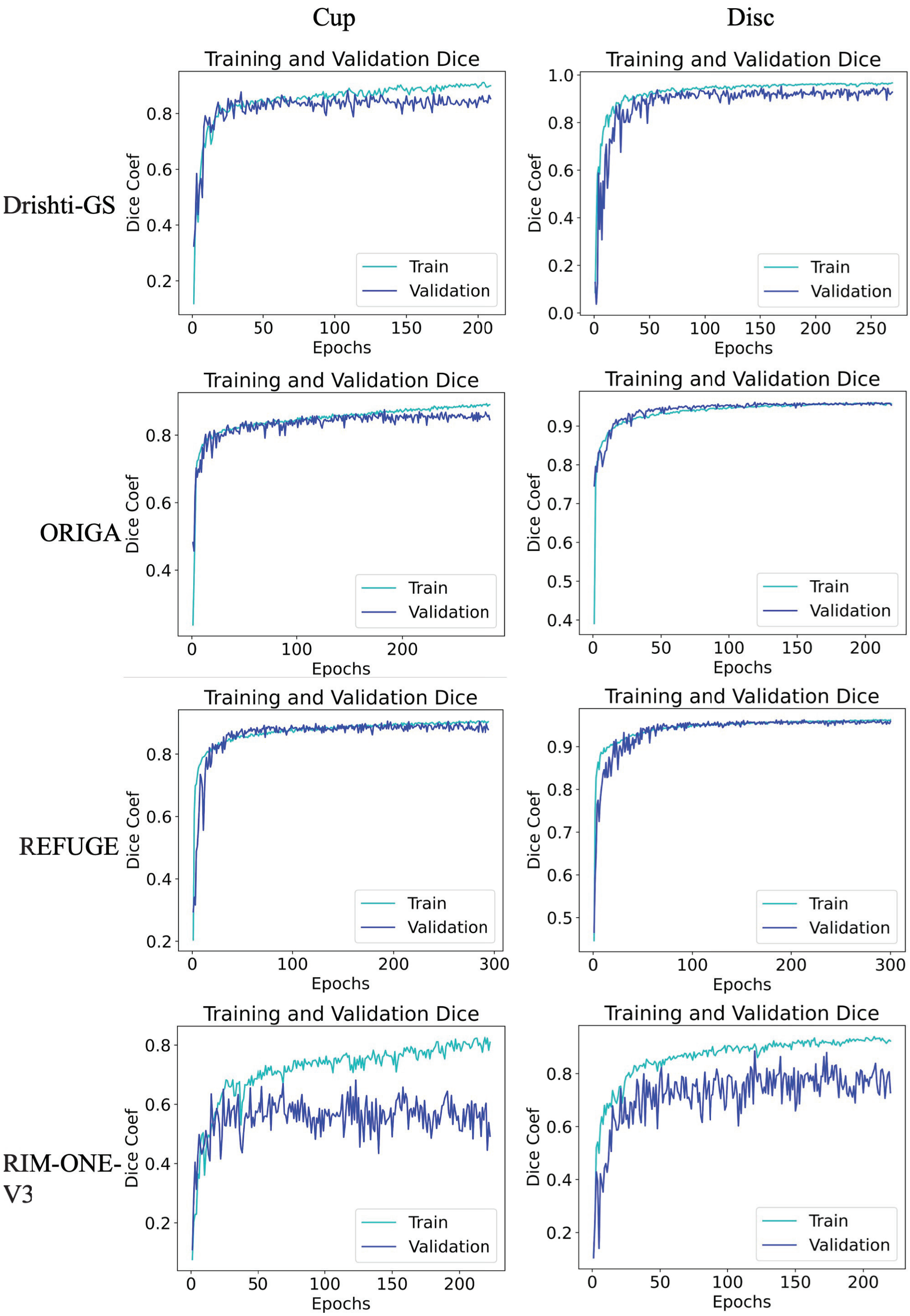


Figure 5. Accuracy (Jaccard Score) function curves on various datasets (epoch vs. Dice coefficient). Cyan is the train accuracy and blue is the validation accuracy.

Table 2. Dice and Jaccard evaluation metrics for various datasets.

Dataset	Optic Disc Segmentation		Optic Cup Segmentation	
	Dice/F1 Score	Jaccard Score/IoU	Dice/F1 Score	Jaccard Score/IoU
Drishti-GS	0.943	0.893	0.889	0.801
RIM-ONE-V3	0.910	0.838	0.649	0.770
ORIGA	0.962	0.928	0.871	0.773
REFUGE	0.965	0.933	0.902	0.824

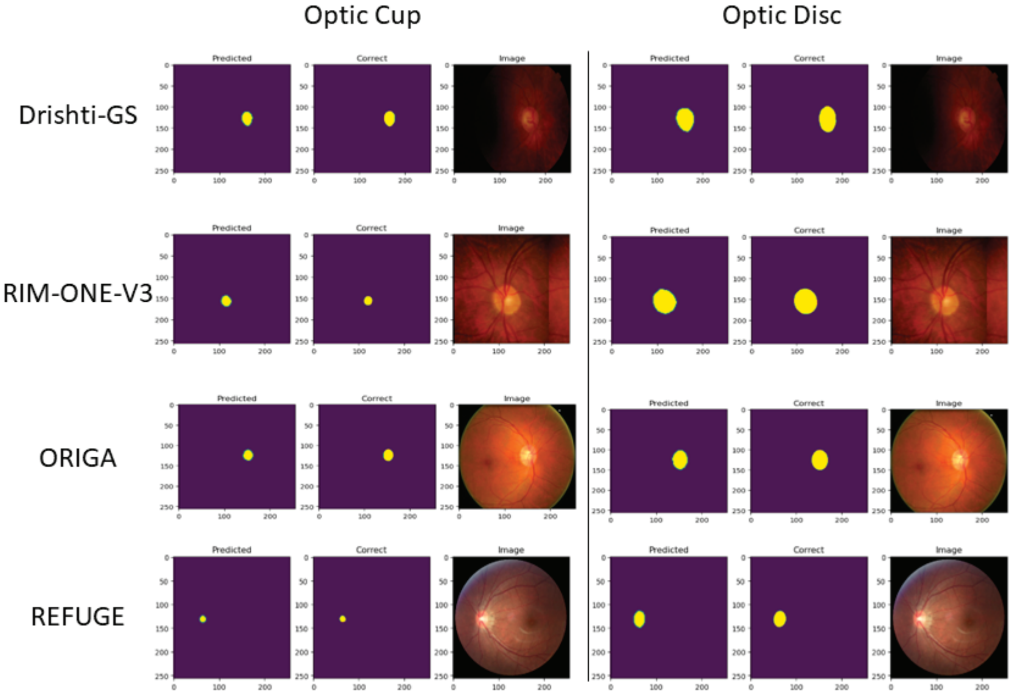


Figure 6. U-Net optic cup and disc segmentations without Canny and dilation for various datasets.

From Table 3, we achieved a ~94% Dice in OD segmentation and an 89% Dice in OC segmentation. In addition, we achieved a 0.89 Jaccard score in OD segmentation and a 0.8 Jaccard score in OC segmentation with the Drishti-GS dataset. Using the RIM-ONE-V3 dataset, we achieved a 91% Dice in OD segmentation and a 64% Dice in OC segmentation. Additionally, we achieved a 0.83 Jaccard for OD segmentation and a 0.77 Jaccard for OC segmentation. Our model achieved an approximate 97% Dice in OD segmentation and a 90% Dice in OC segmentation with the REFUGE dataset. The model also had a 0.93 Jaccard score OD segmentation and a 0.82 Jaccard score in OC segmentation with the REFUGE dataset. Lastly, using the ORIGA dataset, the model delivered a ~96% Dice and an ~87% Dice for OD and OC segmentation, respectively. Additionally, it delivered a 0.928 Jaccard and 0.773 Jaccard for OD and OC segmentation, respectively.

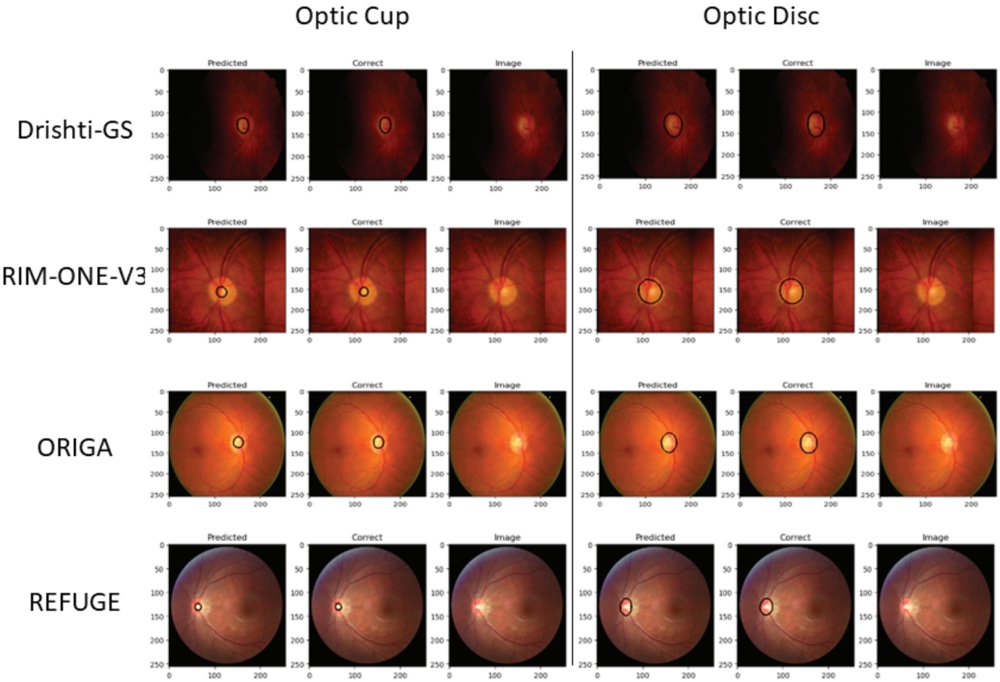


Figure 7. U-Net optic cup and disc segmentations with Canny and dilation for various datasets.

Table 3. OD and OC segmentation results on Drishti-GS and REFUGE datasets.

Datasets	Methods	OD Segmentation		OC Segmentation	
		DC	JAC	DC	JAC
REFUGE	M-Net [12]	0.943	-	0.831	-
	M-Ada [25]	0.958	-	0.882	-
	EARDS [27]	0.954	0.914	0.887	0.801
	pOSAL [48]	0.946	-	0.875	-
	Multi-Model [49]	-	0.922	-	0.790
	CFEA [50]	0.941	-	0.862	-
	Two-Stage Mask R-CNN [51]	0.947	-	0.854	-
	Ours	***0.965***	***0.933***	***0.902***	***0.824***
ORIGA	Deep object detection Network [7]	0.845	-	0.845	-
	JointRCNN [19]	0.937	-	0.794	-
	SS-DCGAN [38]	0.901	-	-	-
	Ours	***0.962***	***0.928***	***0.871***	***0.773***
Drishti-GS	U-Net [3]	0.950	-	0.800	-
	[11]	0.973	0.949	0.887	0.804
	FC-DenseNet [12]	0.949	0.904	0.828	0.711
	M-Net [14]	0.959	-	0.866	-

Table 3. *Cont.*

Datasets	Methods	OD Segmentation		OC Segmentation	
		DC	JAC	DC	JAC
	M-Ada [25]	0.971	-	0.910	-
	EARDS [27]	0.974	0.949	0.915	0.849
	ResFPN-Net [29]	0.976	-	0.896	-
	WRoIM [52]	0.960	-	0.890	-
	WGAN [53]	0.954	-	0.840	-
	pOSAL [48]	0.965	-	0.858	-
	GL-Net [54]	0.971	-	0.905	-
	Multi-Model [49]	0.960	0.924	0.902	0.822
	Ours	***0.943***	***0.893***	***0.889***	***0.801***
RIM-ONE -V3	Hybrid [8]	0.930	0.910	0.910	0.880
	Modified U-Net [9]	0.950	0.890	0.820	0.690
	ECSD [32]	0.860	0.760	0.800	0.680
	EE-U-Net [34]	0.950	0.880	0.860	0.760
	pOSAL [48]	0.860	-	0.787	-
	Ours	***0.910***	***0.830***	***0.640***	***0.770***

5. Discussion

In this section, we start by comparing our U-Net model evaluation to those state-of-the-art approaches referenced in Table 3. While simple, our model performs on par with the presented models in the same datasets and performs slightly worse given Drishti-GS and RIM-ONE-V3 datasets due to the lack of image data. This is true for both the OD and OC segmentation results. On the other hand, our method performs slightly better than the other state-of-the-art models when run over the REFUGE and ORIGA datasets. Table 3 shows the related Dice and Jaccard metrics for both OD and OC segmentation, although most of the models do not run Jaccard for either OD or OC segmentation. Figures 6 and 7 show results that are correlated with the model evaluation in Tables 2 and 3. With the model performance being competitive, our representation of the edge detection and dilation deliver optimal results for CDR analysis for glaucoma. Concerning the results from the Drishti-GS dataset, FC-DenseNet [12] has a similar performance to our model for OD segmentation when we consider the Dice (0.949) and Jaccard (0.904) scores. For the OC segmentation, the multi-model [52] is also very similar in performance in terms of Dice (0.902) and Jaccard (0.822) scores. The results using the RIM-ONE-V3 dataset were comparable for OD segmentations. Both Drishit-GS and RIM-ONE-V3 have very small datasets. Our results for the ORIGA and REFUGE datasets are much higher with none of the existing models being comparable. We speculate that the reason for this is twofold: (1) both the REFUGE and ORIGA datasets have many images (650 and 1200, respectively) to scatter across training, validation and testing; (2) there is a clear color boundary that helps define the optic cup and optic disc, clearly helping the model better distinguish between them. The consistency in the sizes of masks in both datasets is indicative of this.

6. Conclusions

The work outlined herein displayed an end-to-end separative OD and OC segmentation approach. We first employ a modified U-Net encoder to find the features map and then a decoder to upsample the image back. The output is fed into an edge detector that gives a thin boundary around the edge. Dilation is next applied to thicken the edge boundary for a better visual representation. These results indicate that the boundary is accurate and can be subsequently used for analyzing CDR in instances of glaucoma. Based on these results, our model is as competent as the existing state-of-the-art models and performs better using both the ORIGA and REFUGE datasets even with a simple network architecture.

Further work should be directed towards having a CDR detector that optimizes (fixes) our segmentation results according to the correct ratio requirement. Although our model achieves robust results even with a slightly modified U-Net pipeline, it remains to be seen how changing the model and using various backbone training methods would impact performance. This includes the ensemble models that would average pixel classifications for the most accurate detection. It is also possible to first apply object detection to crop the data prior to segmentation. Note that the current studies are limited by the use of 2D data; the prospect of processing 3D fundus images would be an extension that provides spatial data for segmentation, potentially yielding better results because of the greater definition between disc and cup pixels. In summary, we have developed a simple novel architecture that performs as well as, and sometimes better than, existing methods that automate the processing of fundus images for assisting the analysis of CDR in instances of glaucoma.

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

Funding: This project is partially funded by the National Institutes of Health, project number: 1R15GM139115-01 (RC). The Article Processing Charges is supported by the Department of Ophthalmology, University Hospitals Cleveland Medical Center (M.Y). Project Code: https://github.com/SrikanthTad/Glaucoma-Cup-and-Disc-Segmentation.git (accessed on 19 March 2023).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Abramoff, M.D.; Garvin, M.K.; Sonka, M. Retinal imaging and image analysis. *IEEE Rev. Biomed. Eng.* **2010**, *3*, 169–208. [CrossRef] [PubMed]
2. Bourne, R.R. Worldwide glaucoma through the looking glass. *Br. J. Ophthalmol.* **2006**, *90*, 253–254. [CrossRef] [PubMed]
3. Swetha, M.; Chitra Devi, M.; Jayashankari, J.M.E.; Veeralakshmi, P. Automated Diagnosis of Glaucoma Using Cup to Disc Ratio. *JETIR* **2020**, *7*, 278–282.
4. Ronneberger, O.F.; 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; p. 9351.
5. Cheng, J.; Liu, J.; Xu, Y.; Yin, F.; Wong, D.W.K.; Tan, N.-M.; Tao, D.; Cheng, C.-Y.; Aung, T.; Wong, T.Y. Superpixel Classification Based Optic Disc and Optic Cup Segmentation for Glaucoma Screening. *IEEE Trans. Med. Imaging* **2013**, *32*, 1019–1032. [CrossRef]
6. Sarkar, D.C.; Das, S. Automated Glaucoma Detection of Medical Image Using Biogeography Based Optimization. In *Advances in Optical Science and Engineering*; Springer Proceedings in Physics; Bhattacharya, I., Chakrabarti, S., Reehal, H., Lakshminarayanan, V., Eds.; Springer: Singapore, 2017.
7. Sun, X.; Xu, Y.; Tan, M.; Fu, H.; Zhao, W.; You, T.; Liu, J. Localizing Optic Disc and Cup for Glaucoma Screening via Deep Object Detection Networks. In Proceedings of the Computational Pathology and Ophthalmic Medical Image Analysis: First International Workshop, COMPAY 2018, and 5th International Workshop, OMIA 2018, Granada, Spain, 16–20 September 2018.
8. Thakur, N.; Juneja, M. Optic disc and optic cup segmentation from retinal images using hybrid approach. *Expert Syst. Appl.* **2019**, *127*, 308–322. [CrossRef]
9. Sevastopolsky, A. Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network. *Pattern Recognit. Image Anal.* **2017**, *27*, 618–624. [CrossRef]

10. Kim, J.; Tran, L.Q.; Chew, E.Y.; Antani, S.K. Optic Disc and Cup Segmentation for Glaucoma Characterization Using Deep Learning. In Proceedings of the 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), Cordoba, Spain, 5–7 June 2019; pp. 489–494.
11. Yu, S.; Xiao, D.; Frost, S.; Kanagasingam, Y. Robust optic disc and cup segmentation with deep learning for glaucoma detection. *Comput. Med. Imaging Graph. Off. J. Comput. Med. Imaging Soc.* **2019**, *74*, 61–71. [CrossRef]
12. Al-Bander, B.; Williams, B.M.; Al-Nuaimy, W.; Al-Taee, M.A.; Pratt, H.; Zheng, Y. Dense Fully Convolutional Segmentation of the Optic Disc and Cup in Colour Fundus for Glaucoma Diagnosis. *Symmetry* **2018**, *10*, 87. [CrossRef]
13. Guo, F.; Li, W.; Tang, J.; Zou, B.; Fan, Z. Automated glaucoma screening method based on image segmentation and feature extraction. *Med. Biol. Eng. Comput.* **2020**, *58*, 2567–2586. [CrossRef]
14. Fu, H.; Cheng, J.; Xu, Y.; Wong, D.W.K.; Liu, J.; Cao, X. Joint Optic Disc and Cup Segmentation Based on Multi-Label Deep Network and Polar Transformation. *IEEE Trans. Med. Imaging* **2018**, *37*, 1597–1605. [CrossRef]
15. Bajwa, M.N.S.; Singh, G.A.P.; Neumeier, W.; Malik, M.I.; Dengel, A.; Ahmed, S. G1020: A Benchmark Retinal Fundus Image Dataset for Computer-Aided Glaucoma Detection. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020.
16. Sudhan, M.B.; Sinthuja, M.; Pravinth Raja, S.; Amutharaj, J.; Charlyn Pushpa Latha, G.; Sheeba Rachel, S.; Anitha, T.; Rajendran, T.; Waji, Y.A. Segmentation and Classification of Glaucoma Using U-Net with Deep Learning Model. *J. Healthc. Eng.* **2022**, *2022*, 1601354. [CrossRef] [PubMed]
17. Juneja, M.; Singh, S.; Agarwal, N.; Bali, S.; Gupta, S.; Thakur, N.; Jindal, P. Automated detection of Glaucoma using deep learning convolution network (G-net). *Multimed. Tools Appl.* **2020**, *79*, 15531–15553. [CrossRef]
18. Pascal, L.; Perdomo, O.J.; Bost, X.; Huet, B.; Otalora, S.; Zuluaga, M.A. Multi-task deep learning for glaucoma detection from color fundus images. *Sci. Rep.* **2022**, *12*, 12361. [CrossRef] [PubMed]
19. Jiang, Y.; Duan, L.; Cheng, J.; Gu, Z.; Xia, H.; Fu, H.; Li, C.; Liu, J. JointRCNN: A Region-Based Convolutional Neural Network for Optic Disc and Cup Segmentation. *IEEE Trans. Biomed. Eng.* **2020**, *67*, 335–343. [CrossRef] [PubMed]
20. Gu, Z.; Cheng, J.; Fu, H.; Zhou, K.; Hao, H.; Zhao, Y.; Zhang, T.; Gao, S.; Liu, J. CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Trans. Med. Imaging* **2019**, *38*, 2281–2292. [CrossRef]
21. Liu, H.; Feng, Y.; Xu, H.; Liang, S.; Liang, H.; Li, S.; Zhu, J.; Yang, S.; Li, F. MEA-Net: Multilayer edge attention network for medical image segmentation. *Sci. Rep.* **2022**, *12*, 7868. [CrossRef]
22. Bajwa, M.N.; Malik, M.I.; Siddiqui, S.A.; Dengel, A.; Shafait, F.; Neumeier, W.; Ahmed, S. Two-stage framework for optic disc localization and glaucoma classification in retinal fundus images using deep learning. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 136.
23. Xie, Z.; Ling, T.; Yang, Y.; Shu, R.; Liu, B.J. Optic Disc and Cup Image Segmentation Utilizing Contour-Based Transformation and Sequence Labeling Networks. *J. Med. Syst.* **2020**, *44*, 96. [CrossRef]
24. Gao, J.; Jiang, Y.; Zhang, H.; Wang, F. Joint disc and cup segmentation based on recurrent fully convolutional network. *PLoS ONE* **2020**, *15*, e0238983. [CrossRef]
25. Hervella, Á.S.; Rouco, J.; Novo, J.; Ortega, M. End-to-end multi-task learning for simultaneous optic disc and cup segmentation and glaucoma classification in eye fundus images. *Appl. Soft Comput.* **2021**, *116*, 108347. [CrossRef]
26. Parkhi, P.; Hambarde, B.H. Optical Cup and Disc Segmentation using Deep Learning Technique for Glaucoma Detection. *Int. J. Next Gener. Comput.* **2023**, *14*, 44–52. [CrossRef]
27. Zhou, W.; Ji, J.; Jiang, Y.; Wang, J.; Qi, Q.; Yi, Y. EARDS: EfficientNet and attention-based residual depth-wise separable convolution for joint OD and OC segmentation. *Front. Neurosci.* **2023**, *17*, 1139181. [CrossRef] [PubMed]
28. Wu, J.; Fu, R.; Fang, H.; Zhang, Y.; Xu, Y. MedSegDiff-V2: Diffusion based Medical Image Segmentation with Transformer. *arXiv* **2023**, arXiv:2301.11798.
29. Sun, G.; Zhang, Z.; Zhang, J.; Zhu, M.; Zhu, X.; Yang, J.; Li, Y. Joint optic disc and cup segmentation based on multi-scale feature analysis and attention pyramid architecture for glaucoma screening. *Neural Comput. Appl.* **2021**. [CrossRef]
30. Xue, X.; Wang, L.; Du, W.; Fujiwara, Y.; Peng, Y. Multiple Preprocessing Hybrid Level Set Model for Optic Disc Segmentation in Fundus Images. *Sensors* **2022**, *22*, 6899. [CrossRef]
31. Zaaboub, N.; Sandid, F.; Douik, A.; Solaiman, B. Optic disc detection and segmentation using saliency mask in retinal fundus images. *Comput. Biol. Med.* **2022**, *150*, 106067. [CrossRef]
32. Liu, B.; Pan, D.; Shuai, Z.; Song, H. ECSD-Net: A joint optic disc and cup segmentation and glaucoma classification network based on unsupervised domain adaptation. *Comput. Methods Programs Biomed.* **2022**, *213*, 106530. [CrossRef]
33. Xiong, H.; Liu, S.; Sharan, R.V.; Coiera, E.; Berkovsky, S. Weak label based Bayesian U-Net for optic disc segmentation in fundus images. *Artif. Intell. Med.* **2022**, *126*, 102261. [CrossRef]
34. Wang, J.; Li, X.; Cheng, Y. Towards an extended EfficientNet-based U-Net framework for joint optic disc and cup segmentation in the fundus image. *Biomed. Signal Process. Control* **2023**, *85*, 104906. [CrossRef]
35. Sivaswamy, J.; Krishnadas, S.R.; Joshi, G.D.; Jain, M.; Tabish, A.U. Drishti-gs: Retinal image dataset for optic nerve head(onh) segmentation. In Proceedings of the 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI), Beijing, China, 29 April–2 May 2014; pp. 53–56. [CrossRef]
36. Sivaswamy, J.; Krishnadas, S.R.; Chakravarty, A.; Joshi, G.D.; Tabish, A.U. A comprehensive retinal image dataset for the assessment of glaucoma from the optic nerve head analysis. *JSM Biomed. Imaging Data Pap.* **2015**, *2*, 1–7.

37. Diaz-Pinto, A.; Morales, S.; Naranjo, V.; Köhler, T.; Mossi, J.M.; Navea, A. CNNs for automatic glaucoma assessment using fundus images: An extensive validation. *Biomed. Eng. Online* **2019**, *18*, 29. [CrossRef] [PubMed]
38. Diaz-Pinto, A.; Colomer, A.; Naranjo, V.; Morales, S.; Xu, Y.; Frangi, A.F. Retinal Image Synthesis and Semi-Supervised Learning for Glaucoma Assessment. *IEEE Trans. Med. Imaging* **2019**, *38*, 2211–2218. [CrossRef] [PubMed]
39. Zilly, J.; Buhmann, J.M.; Mahapatra, D. Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation. *Comput. Med. Imaging Graph. Off. J. Comput. Med. Imaging Soc.* **2017**, *55*, 28–41. [CrossRef]
40. Phasuk, S.; Tantibundhit, C.; Poopresert, P.; Yaemsuk, A.; Suvannachart, P.; Itthipanichpong, R.; Chansangpetch, S.; Manassakorn, A.; Tantisevi, V.; Rojanapongpun, P. Automated Glaucoma Screening from Retinal Fundus Image Using Deep Learning. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Berlin, Germany, 23–27 July 2019; pp. 904–907. [CrossRef]
41. Wang, J.; Yan, Y.; Xu, Y.; Zhao, W.; Min, H.; Tan, M.; Liu, J. Conditional Adversarial Transfer for Glaucoma Diagnosis. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Berlin, Germany, 23–27 July 2019; pp. 2032–2035. [CrossRef]
42. Chen, X.; Xu, Y.; Wong, D.W.K.; Wong, T.Y.; Liu, J. Glaucoma detection based on deep convolutional neural network. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Milan, Italy, 25–29 August 2015; pp. 715–718. [CrossRef]
43. Li, A.; Cheng, J.; Wong, D.W.K.; Liu, J. Integrating holistic and local deep features for glaucoma classification. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 August 2016; pp. 1328–1331. [CrossRef]
44. Fumero, F.; Sigut, J.F.; Alayón, S.; Gonzalez-Hernandez, M.; Rosa, M.G. Interactive Tool and Database for Optic Disc and Cup Segmentation of Stereo and Monocular Retinal Fundus Images. In Proceedings of the 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2015 in Co-Operation with EUROGRAPHICS, Pilsen, Czech Republic, 8–12 June 2015; pp. 91–97. Available online: http://wscg.zcu.cz/DL/wscg_DL.htm (accessed on 19 March 2023).
45. Cerentini, A.; Welfer, D.; d'Ornellas, M.C.; Pereira Haygert, C.J.; Dotto, G.N. Automatic Identification of Glaucoma Using Deep Learning Methods. In *MEDINFO 2017: Precision Healthcare Through Informatics: Proceedings of the 16th World Congress on Medical and Health Informatics*; Studies in Health Technology and Informatics; IOS Press: Amsterdam, The Netherlands, 2017; Volume 245, pp. 318–321.
46. Orlando, J.I.; Fu, H.; Breda, J.B.; van Keer, K.; Bathula, D.R.; Diaz-Pinto, A.; Fang, R.; Heng, P.-A.; Kim, J.; Lee, J.; et al. Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. *Med. Image Anal.* **2020**, *59*, 101570. [CrossRef] [PubMed]
47. Ding, L.; Goshtasby, A. On the Canny edge detector. *Pattern Recognit.* **2001**, *34*, 721–725. [CrossRef]
48. Wang, S.; Yu, L.; Yang, X.; Fu, C.W.; Heng, P.A. Patch-Based Output Space Adversarial Learning for Joint Optic Disc and Cup Segmentation. *IEEE Trans. Med. Imaging* **2019**, *38*, 2485–2495. [CrossRef]
49. Hervella, Á.S.; Ramos, L.; Rouco, J.; Novo, J.; Ortega, M. Multi-Modal Self-Supervised Pre-Training for Joint Optic Disc and Cup Segmentation in Eye Fundus Images. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 961–965.
50. Liu, P.; Kong, B.; Li, Z.; Zhang, S.; Fang, R. CFEA: Collaborative Feature Ensembling Adaptation for Domain Adaptation in Unsupervised Optic Disc and Cup Segmentation. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2019*; MICCAI 2019. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11768. [CrossRef]
51. Almubarak, H.; Bazi, Y.; Alajlan, N. Two-Stage Mask-RCNN Approach for Detecting and Segmenting the Optic Nerve Head, Optic Disc, and Optic Cup in Fundus Images. *Appl. Sci.* **2020**, *10*, 3833. [CrossRef]
52. Shah, S.; Kasukurthi, N.; Pande, H. Dynamic region proposal networks for semantic segmentation in automated glaucoma screening. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 578–582. [CrossRef]
53. Kadambi, S.; Wang, Z.; Xing, E. WGAN domain adaptation for the joint optic disc-and-cup segmentation in fundus images. *Int. J. Comput. Assist. Radiol. Surg.* **2020**, *15*, 1205–1213. [CrossRef]
54. Jiang, Y.; Tan, N.; Peng, T. Optic Disc and Cup Segmentation Based on Deep Convolutional Generative Adversarial Networks. *IEEE Access* **2019**, *7*, 64483–64493. [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

A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features

Hareem Kibriya [1], Rashid Amin [2], Jinsul Kim [3,*], Marriam Nawaz [4] and Rahma Gantassi [5]

[1] Department of Computer Sciences, University of Engineering and Technology, Taxila 47050, Pakistan
[2] Department of Computer Sciences, University of Chakwal, Chakwal 48800, Pakistan
[3] School of Electronics and Computer Engineering, Chonnam National University, 300 Yongbong-dong, Buk-gu, Gwangju 500757, Republic of Korea
[4] Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan
[5] Department of Electrical Engineering, Chonnam National University, Gwangju 61186, Republic of Korea
* Correspondence: jsworld@jnu.ac.kr; Tel.: +82-10-2914-0001

Abstract: One of the most severe types of cancer caused by the uncontrollable proliferation of brain cells inside the skull is brain tumors. Hence, a fast and accurate tumor detection method is critical for the patient's health. Many automated artificial intelligence (AI) methods have recently been developed to diagnose tumors. These approaches, however, result in poor performance; hence, there is a need for an efficient technique to perform precise diagnoses. This paper suggests a novel approach for brain tumor detection via an ensemble of deep and hand-crafted feature vectors (FV). The novel FV is an ensemble of hand-crafted features based on the GLCM (gray level co-occurrence matrix) and in-depth features based on VGG16. The novel FV contains robust features compared to independent vectors, which improve the suggested method's discriminating capabilities. The proposed FV is then classified using SVM or support vector machines and the k-nearest neighbor classifier (KNN). The framework achieved the highest accuracy of 99% on the ensemble FV. The results indicate the reliability and efficacy of the proposed methodology; hence, radiologists can use it to detect brain tumors through MRI (magnetic resonance imaging). The results show the robustness of the proposed method and can be deployed in the real environment to detect brain tumors from MRI images accurately. In addition, the performance of our model was validated via cross-tabulated data.

Keywords: brain tumor; artificial intelligence; GLCM; KNN; VGG16

Citation: Kibriya, H.; Amin, R.; Kim, J.; Nawaz, M.; Gantassi, R. A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features. *Sensors* **2023**, *23*, 4693. https://doi.org/10.3390/s23104693

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 7 March 2023
Revised: 24 April 2023
Accepted: 8 May 2023
Published: 12 May 2023

1. Introduction

The brain is an important organ that oversees entire bodily functions and is a vital organ that is a central part of the nervous system. It can be affected by one of the most lethal brain diseases, called a brain tumor, caused by an unusual growth of cell proliferation within the brain. These tumors cause brain damage and dysfunctions, which can be very lethal if left untreated for a long time. The World Health Organization (WHO) has predicted an annual 5% increase in brain tumors [1]. Another report declares brain tumors the 10th leading cause of mortality among humans. According to estimates, at least 18,600 individuals will die from a deadly brain or central nervous system (CNS) tumor this year [2]. Thus, on-time and accurate tumor diagnosis can increase the patient's chances of survival.

The patient's health depends on a prompt and precise diagnosis of a brain tumor because the stage and kind of tumor influence the course of treatment. As tumors differ in size, location, and shape, identifying one can be challenging. An inaccurate or delayed diagnosis may lower the likelihood of a patient surviving a brain tumor. Medical professionals have historically used visual evaluation of medical imaging and precise tumor location tracing to diagnose brain tumors. These medical images are from MRIs, computed tomography (CT), and positron emission tomography (PET). Radiologists and medical experts frequently utilize MRI scans to identify brain tumors because they produce high-quality images of soft tissues [3]. Because of the surrounding healthy tissues, the tumor margins are frequently hazy when manually detecting tumors through optical inspection. Because of this, manually identifying tumors takes a long time and usually results in incorrect tumor diagnosis. Images that are noisy for various reasons, such as medical image acquisition techniques or variations in imaging equipment, are another cause of tumor misidentification [4]. Usually, biopsies are carried out to ascertain whether the tissue is malignant or benign. Determining the cancer is a painful and time-consuming technique. Consequently, automated technologies are replacing traditional approaches due to the complexity of current approaches [5–7].

Recognizing and categorizing brain tumors early for a patient to receive the proper treatment is crucial. Thanks to technological advancements, professionals may treat patients properly using automated healthcare systems. More research is being presented to address the challenges in medical image identification as machine learning and artificial intelligence technologies improve [8]. These automated methods effectively diagnose brain tumors and aid medical personnel in deciding appropriate treatment procedures [9]. Given that the human eye cannot distinguish between the numerous shades of grey in MRI images, these systems are extremely useful for detecting even the slightest color changes. A persistent problem in medical image analysis is storing and evaluating large amounts of medical data. However, the existing systems cannot effectively manage the noticeable increase in data volume in the medical sector. Currently, contemporary machine learning algorithms frequently utilize big data approaches to analyze medical image data. The development of new technologies, especially machine learning and artificial intelligence, has substantially impacted the medical industry because it has provided medical departments with an essential tool to get second opinions with much higher precision. The robust automated frameworks work best where radiologists want to minimize the possibility of biopsy or inspect tumor depth or type [10].

Although numerous works have been proposed for the effective recognition of brain tumors, there is room for performance improvement. Employing only deep learning algorithms demands vast quantities of labeled data, while the supervised approach to classifying a brain tumor has much potential; however, it takes specialized knowledge to extract the best characteristics and selection methods. The paper proposes a novel and fully automated brain tumor identification and classification method that combines hand-engineered and deep features to fill this gap. The novel FV is an ensemble of hand-crafted features based on the GLCM (gray level co-occurrence matrix) and deep features based on VGG16. These FVs, when combined, increase the discriminating ability of the proposed system, allowing for accurate tumor detection even in the presence of backgrounds, ill-defined tumor boundaries, skulls, and other MRI artifacts. The proposed FV is then classified using SVM or support vector machines and the k-nearest neighbor classifier (KNN). The suggested framework performs better than the existing methods at quickly and accurately detecting brain cancers. The following are the primary contributions of the suggested system:

- We present a novel method based on an ensemble of deep and hand-crafted features to classify brain tumors in MR images.
- Per our knowledge, this is the first-ever study based on a feature-level ensemble of VGG16 and GLCM features to classify brain tumors.
- Our framework consists of three main core steps: deep feature extraction via CNN, that is, through the VGG16 model, hand-crafted feature computation via GLCM, creating an ensemble vector of these FVs, and finally, classification using SVM and KNN.
- The proposed method effectively classifies brain tumors because the fusion of the GLCM and deep FV computes an effective set of image features, resulting in better discrimination of tumor and normal images.
- The results indicate the efficacy of the presented approach as compared to existing methodologies.

2. Related Work

Early tumor detection and classification of brain tumors are required for efficient patient therapy. Thanks to impressive technological advancements, specialists may treat patients more effectively using automated healthcare systems. Researchers have recently developed several methods for classifying brain tumors that use ML and DL-based algorithms. New artificial intelligence and deep learning technologies have substantially impacted medical picture processing, notably in an illness diagnosis. This section examines the research on methods for classifying brain tumors using ML and DL-based algorithms. Anaraki et al. [7] used Genetic Algorithms to perform brain tumor categorization in the pituitary, glioma, and meningioma malignancies from MRI images. The system attained 94.2% accuracy. However, the algorithm could not identify an ideal CNN design, leading to poor performance. Using the complete volumetric T1-Gado MRI sequence from the Brats 2018 dataset, Mzoughi et al. [11] presented DL architectures to grade glioma tumors based on severity. The authors, in their framework, incorporated local and global features with lower weights by using small kernels that resulted in an accuracy score of 96%. Sejuti et al. [12] presented a CNN-SVM-based method to identify and classify brain tumors in MRI images and attained 97.1% accuracy. Whereas the authors of Abiwinanda et al. [13] designed five different CNN frameworks to detect tumors in the brain. The system obtained 84.1% accuracy. However, it is worth mentioning that relatively basic CNNs cannot extract complex high-level features, leading to mediocre overall accuracy. Due to this, the CNNs in [11–13] have resulted in poor performance because of very simple architectural designs.

To identify an ideal CNN design with lower computation costs for the classification of brain tumors, a hierarchical deep learning-based brain tumor classification technique was presented by Khan et al. [14]. The study obtained 92% accuracy on the Kaggle database. However, the system must be evaluated rigorously to view its efficacy in real-case scenarios. Alanazi et al. [15] demonstrated a brain tumor detection system composed of 22 layers, resulting in 96.8% accuracy. However, only a few imaging samples are used to conduct the study; hence, a thorough evaluation is crucial. Afshar et al. [16] used capsule networks that achieved an accuracy of 90.8%. However, one of the limitations of capsule networks is their sensitivity to image backgrounds. These architectures tend to perform much better in the case of segmentation image input. Noureen et al. [17] fine-tuned the Inception-v3 and Xception models for deep CNN features extraction, whereas an ensemble of different classifiers, i.e., SVM, KNN, and random forest (RF), was used to classify the images. The system obtained 93.3% overall accuracy. In another study, the authors of Swati et al. [6] trained and evaluated transfer-learned AlexNet architecture to identify brain tumors from MR images, resulting in 89.9% overall accuracy. However, due to low overall accuracy, the systems in [6,17] must be evaluated before deploying in real-world scenarios.

Kang et al. [18] presented a feature ensemble method using vectors obtained from DenseNet169, Inception-v3, and ResNeXt50. The system resulted in 98.5% accuracy. In another study, the authors of Waghmere et al. [19] developed a brain tumor classification method using VGG16 architecture. The system resulted in 95.7% accuracy on preprocessed and augmented MRI images. However, the studies mentioned in [18,19] are computationally complex. Some of the authors also proposed segmentation before classification methods. For instance, . Naser et al. [20] employed a U-Net architecture to segment the tumors and VGG16 to perform classification. The study was performed on the TCIA dataset and achieved almost 92% accuracy. Masood et al. [21] employed Mask RCNN to classify brain tumors. In the first stage, they localized the tumor region using bounding boxes, and in the next stage, they classified the tumor. A multimodal tumor classification system built on CNN was developed by Sajjad et al. [22]. They used input cascade CNN to first segment the MRIs, then classified them with 94.5% accuracy using a tuned VGG-19. However, the systems are computationally expensive as they perform segmentation before the classification phase.

On the other hand, various researchers have proposed ML-based algorithms to detect brain tumors using hand-crafted feature extraction and classification. Amin et al. [23]

extracted and classified hand-crafted features using GLCM and SVM, respectively, and obtained 97% accuracy. Kaplan et al. [24] used Local Binary Pattern (LBP) for feature extraction KNN for classification. They achieved 95.5% overall accuracy. Bahadure et al. [25] segmented the tumor region and extracted hand-crafted features using GLCM. The system obtained 96.5% accuracy via SVM. Garg et al. [26] applied the Otsu threshold to MRI images and extracted GLCM features. They performed classification via SVM and KNN with 97.3% accuracy. Minz et al. [27] extracted GLCM features from segmented images and performed classification via AdaBoost. The system obtained the highest accuracy of 89.9%. However, these systems require manual segmentation of tumor regions before the feature extraction and classification phase, thus increasing the system's complexity. Raja et al. [28] used information-theoretic measures and Bayesian fuzzy clustering techniques to segment images, whereas the nonlocal mean filter for image de-noising and scattering transform. They used Tsallis entropy as feature extractors. Finally, a hybrid DAE approach was deployed for the classification of tumors. However, the computation required for this method is time-consuming and inefficient. Ali et al. [29] proposed an approach by employing two deep learning approaches named the GoogleNet and YOLO models to recognize the brain tumors from the MRI samples. They attained the highest accuracy of 97% with the first model. Another approach was proposed in [30] that linked RoB and different AI-based architectural clusters in various DL frameworks to investigate them and attain improved analysis results.

In brain tumor segmentation, the k-nearest neighbor algorithm is an object classification technique using the closest learning instances in the problem space. However, KNN is one type of instance-based learning, or lazy learning, in which the function is only locally approximated, and all computations are deferred until classification. The k-nearest neighbor algorithm is one of the simplest machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object assigned to the most frequent class among its k-nearest neighbors. The k-NN algorithm functions as follows:

- Compute the Euclidean or Mahalanobis distance between the target and sampled plots.
- Arrange samples according to the calculated distances.
- Select the optimal k-nearest neighbors heuristically based on the RMSE obtained by the cross-validation technique.
- Compute a weighted average of the inverse distance to the k multivariate nearest neighbors.

The support vector machine was selected because it is an interesting framework from a machine-learning perspective. Specifically, SVM is a linear or non-linear classifier, a mathematical function capable of distinguishing two different types of objects. Such objects are divided into classes, which should not be confused with an implementation [23–27]. Ref. [31] proposed the classification of a brain tumor in brain MRI images using an image mining technique. The median filtering and features preprocessed the MRI images have been extracted using the texture feature extraction technique. Decision tree classification and the interclass relationship in text classification are used to improve the efficiency of traditional mining methods. The system used an SVM classifier, which gives 83% accuracy. Classification of brain tissues through MRI using a hybrid approach of GA and SVM is proposed by Ahmed Kharrat et al. [32]. The features are extracted by the spatial gray level dependence method called SGLDM. The proposed system gives a good accuracy of about 85.22% [33], and proposed morphological operations to the image, then extracted the features. Analysis result from MLPNN and SVM shows these operations can improve classification results in symmetry and grayscale features but reduce results in texture features. Using SVM, the system gets a better result than MLPNN and RBFNN. Because of the brain's symmetrical structure, symmetrical features have better accuracy, and texture features have lower accuracy.

3. Proposed Methodology

This section discusses the proposed brain tumor classification framework in detail. The proposed methodology consists of 4 main steps (as shown in Figure 1): image preprocessing, deep and hand-crafted feature extraction, feature level ensemble of extracted FVs, and finally, tumor classification. Initially, the dataset images are scaled to 225 × 225 to match CNN's input layer size. We extracted hand-engineered and deep features in the feature extraction phase using GLCM and VGG-16, respectively. We presented a feature ensemble methodology that combines both FVs (hand-crafted + deep), which is then classified via SVM and KNN. Feature ensemble combines features from multiple architectures in a single FV, thus eliminating the requirement to use a single FV obtained from low performing model. Feature ensemble leads to better classification outcomes as the new FV is more elaborate and informative than a single one. This method can, thus, aid in developing an efficient model for precisely detecting and classifying brain tumors, which radiologists can use to get a second opinion.

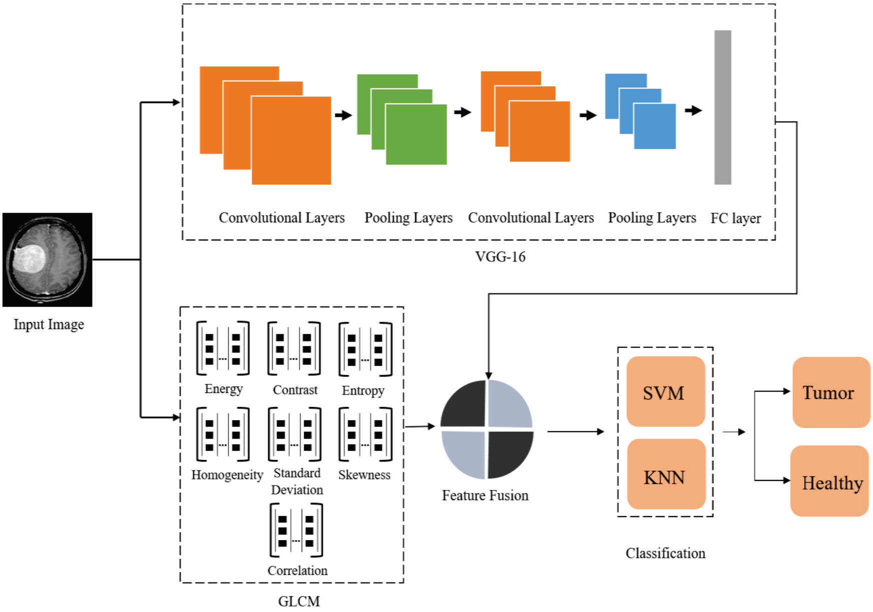


Figure 1. Block diagram of the proposed method indicating all the steps in detail.

3.1. Feature Extraction

Feature extraction collects shape, texture, and color-related information from an image vital for representation. Obtaining optimal features from MRI is a very challenging task [34]. Feature extraction converts raw data into numerical representations while retaining the original information, which is then processed. These features can be extracted using automated techniques (i.e., DL) or manual methods (i.e., hand-engineered feature extraction). However, manual feature extraction requires a detailed understanding of extracting only those significant features for a certain issue. A firm understanding of the context or domain can frequently help decide which attributes might be beneficial. Scientists and researchers have spent years researching strategies for extracting and selecting optimal features from images, signals, audio/video, or text.

On the contrary, DL frameworks automate the feature extraction process without human intervention [35]. The proposed framework extracts hand-crafted features via GLCM and deep CNN features from the fully connected (FC7) layer of VGG16 architecture.

We extracted both hand-engineered and deep features, which are later fused to form a robust, more discriminative feature vector than the independent vector. The details of extracted feature vectors are provided below:

3.1.1. Deep Feature Extraction Using VGG-16

Deep learning is a subset of ML that uses multiple layers of neurons with complex architecture or non-linear processes to simulate high-level data abstractions. With the expansion in data volume and computational capacity, neural networks with more complicated topologies have received attention from academia to industry. Application-wise, deep learning has made significant strides in voice and image categorization, advancing artificial intelligence and human-computer interaction [36].

One of the widely used DL frameworks in a convolutional neural network (CNN) is a feed-forward ANN that employs convolutions in at least one of its layers. It drew inspiration from biological brain networks. CNNs combine ANN with discrete convolutions to extract robust features from the database. These architectures are very effective in identifying and classifying 2D data, i.e., images and videos. These networks take input directly from data (usually images/videos) and automate feature extraction/classification phases, thus saving time compared to conventional image recognition methods. Moreover, exploring robust deep features is crucial in precisely and accurately identifying images (including brain tumors). Due to their benefits, CNNs are known to achieve a state-of-the-art performance when used as deep feature extractors because of the capability to notice minute changes in images and capturing in the form of features [37,38].

In this paper, we used VGG-Net as our deep feature extractor. The framework was presented in 2014 by Karen Simonyan and Andrew Zisserman. The architecture comprised 138 million parameters and was ranked second in ILSVRC in 2014 [39]. The framework is widely employed in image classification due to its robust feature extraction and lightweight architecture [39]. The architecture of VGG-16 is illustrated in Figure 2. The first layer of the framework holds an input image of size 224 × 224. Next to the input layer is a convolution layer (CL) composed of different convolution filters responsible for convolving input images with kernels to obtain feature maps. Multiple stacked-up CLs improve the ability to learn hidden features. CLs are usually followed by an activation function (usually ReLU) that is applied to saturate the generated output. The next layer in this architecture is the pooling layer (PL) which aims to reduce the size of the feature maps to decrease the computational load to the next stages. The FC layer is responsible for producing the class scores from the activations for classification. The VGG-16 architecture comprises 13 CLs with multiple filters of size 3 × 3 throughout the entire network and 5 max-pooling layers. The last three FC layers result in 4096, 4096, and 1000 features [40–42]. The motivation behind using VGG-16 is that the CLs have a kernel size of 3 × 3 with a stride of 1 throughout the network, unlike other CNN models (such as ZF-Net, AlexNet) with a kernel size of 7 × 7 and 11 × 11 with 4 to 5 strides in the initial layers. Usually, the larger strides ignore important patterns in the MR images, whereas a larger kernel size increases the number of parameters.

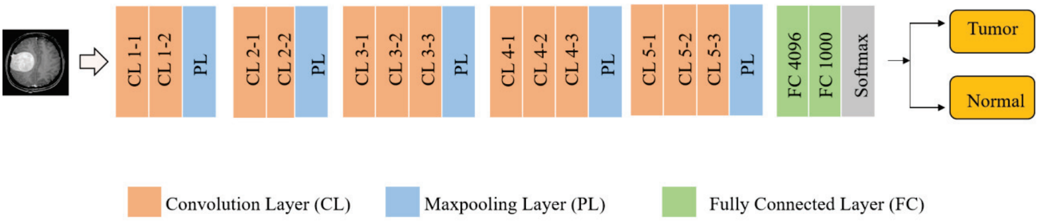


Figure 2. Architectural details of the VGG16 model.

3.1.2. Hand-Crafted Feature Extraction Using GLCM

We also extracted hand-engineered features using GLCM. The main difference between deep features and hand-engineered feature vectors is that hand-crafted features are created in advance by humans and extract a specific set of desired features. In contrast, architecture learns CNN-based features from the provided data [43]. It may be noted that we extracted texture-based hand-crafted features (GLCM) that examine the spatial correlation of pixels among each other. These features are robust in determining a slight change in pixel intensities and can precisely differentiate between tumorous tissues and healthy tissues.

Harlick or GLCM [44] are texture-based features that extract second-order statistical texture features from gray-level images. These features focus on a specific position of the pixels relative to the other pixel, as the image texture usually contains important information regarding the structural arrangement of surfaces. GLCM is proven to produce robust results in different fields, including medical image diagnosis [45], as they can help diagnose brain tumors from MRIs by detecting minor changes in pixel variations. We extracted 7 textural features from MRI images using GLCM: energy, contrast, entropy, homogeneity, shade, and prominence. The hand-crafted features are discussed in this section. We extracted energy which measures the number of pixel repetitions in an image. When the image is homogenous, energy values will be higher, indicating textural uniformity. Energy is computed as in Equation (1). Here m and n denote the size of the GLCM matrix, whereas pixels are represented by x, y

$$Energy = \sqrt{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1} f^2(x,y)} \tag{1}$$

Another feature extracted from the GLCM matrix is contrast, which measures the intensity variation of a pixel and its neighboring pixels in an image. The contrast value will be higher when there is a large variation in image pixels. It can be defined as in Equation (2). Here GLCM matrix is denoted by m and n, whereas image pixels are represented by x and y.

$$Contrast = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1}(x-y)^2 f(x,y) \tag{2}$$

We also calculated entropy which measures the randomness of pixel values in a textural image. Homogenous images possess a higher entropy value than inhomogeneous images, which have a lower entropy value. Entropy can be calculated as in Equation (3). The pixel intensities are represented by x and y, whereas m and n indicate the GLCM matrix.

$$Entropy = -\sum_{x=0}^{m-1}\sum_{y=0}^{n-1} f(x,y) log_2 f(x,y) \tag{3}$$

Inverse difference moment (IDM) determines the similarity and closeness between the pixels of an image to check if the image is textured or non-textured. The IDM's value will be higher in the homogenous images than in non-homogenous images. It can be calculated as in Equation (4)

$$IDM = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\frac{1}{1+(x-y)^2} f(x,y) \tag{4}$$

Standard deviation (SD) describes probability distribution in an observed population. A higher SD value signifies better contrast and intensity in an image. It can be defined as in Equation (5). Here x and y denote pixel values of MRI images, whereas m and n determine the GLCM matrix size.

$$SD = \sqrt{\left(\frac{1}{mxn}\right)\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}(f(x,y) - M^2)} \tag{5}$$

Skewness measures symmetry between image pixels. It will possess lower values when an image is less asymmetric. It is calculated in Equation (6). Pixel intensities are depicted using *x* and *y*. In contrast, the GLCM matrix is denoted by *m*, *n*.

$$Skewness = \left(\frac{1}{mxn}\right)\frac{\sum\left(f(x,y) - M^3\right.}{\partial^3} \tag{6}$$

Correlation defines the spatial dependencies or relations between a pixel and its neighborhood. It can be calculated as in Equation (7), where *x* and *y* denote pixel intensities, whereas *m* and *n* show the size of the GLCM matrix.

$$Correlation = \frac{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}(x,y)f(x,y) - M_x M_y}{\partial_x \partial_y} \tag{7}$$

3.2. Feature Ensemble

Because the performance of an ML classifier is strongly dependent on the input FV, establishing an approach to create a discriminative FV from MRIs is critical for effective tumor classification. Ensemble learning improves the system's performance by merging several FVs in a single predictive FV. It also helps avoid the risk of utilizing an individual FV obtained from one model with poor performance. Depending on the integration, ensemble learning falls into two specific categories, i.e., feature-level ensemble and classifier-level ensemble. At the classifier level, ensemble integration is performed on output sets obtained from classifiers using voting methods to determine the final result. On the contrary, in a feature-level ensemble, the FVs are integrated and fed to the classifiers for the final result. This type of integration yields far better results since the FVs obtained from varying architectures are concatenated, which is much more informative than individual ones because of various features that determine boundaries, edges, shapes, and changes in intensities. Hence, we used this study's feature-level ensemble method to combine deep and hand-engineered FVs. Independent FV acquired via GLCM is defined in Equation (8) which shows the 7 textural features extracted from MRI images, whereas Equation (9) shows 4096 deep features obtained from MRI images using VGG16. These features are then integrated into a novel FV mathematically given in Equation (10). Equation (10) shows the concatenation of both feature vectors computed with the GLCM, and VGG16 models, which results in a novel FV called ensemble vector (EV) comprised of 4103 features representing 2 classes, i.e., normal and tumor.

$$FV_{GLCM\ (2\times 7)} = (GLCM_{2\times 1},\ GLCM_{2\times 2}, \ldots, GLCM_{2\times 7}) \tag{8}$$

$$FV_{VGG16(2\times 4096)} = (VGG16_{2\times 1}\ \ , VGG16_{2\times 2}, \ldots\ VGG16_{2\times 4096}) \tag{9}$$

$$EV_{2x4103} = \sum_{i=1}^{2}(FV_{VGG16},\ FV_{GLCM}\ ) \tag{10}$$

3.3. Classification

Finally, EV is classified via ML-based classifiers, i.e., KNN and SVM. SVM belongs to the family of generalized linear classifiers that avoid overfitting data by maximizing its performance. It is a supervised learning approach developed by Vladimir Wapnik in 1992 [46]. KNN, on the contrary, was developed by Thomas Cover, a supervised learning algorithm deployed in problems related to regression and classification [47]. The algorithm is usually helpful in scenarios with little or no prior knowledge regarding data distribution. The classifier computes the distance between an individual training sample with specified (k) training samples and votes for the most frequent label in those *k* samples [48]. Both classifiers are adopted in classification and regression tasks because of promising results.

4. Results

4.1. Dataset

We conducted this study using two publicly available datasets of brain MR images for brain tumor classification. Both datasets are composed of normal and tumor MRIs. For ease, we named the first dataset BT-small [49] to conduct Study I and the second dataset BT-large to conduct Study II. The BT-small dataset comprises 253 MRI images, of which 155 are tumorous while the rest of the 98 images are normal. BT-large [50] is composed of 1500 normal images and 1500 tumor images. The sample images obtained from both datasets are displayed in Figure 3. We used 70% of the MRI samples to train the models and the remaining 30% of the samples for validation. Table 1 discusses the dataset in detail.

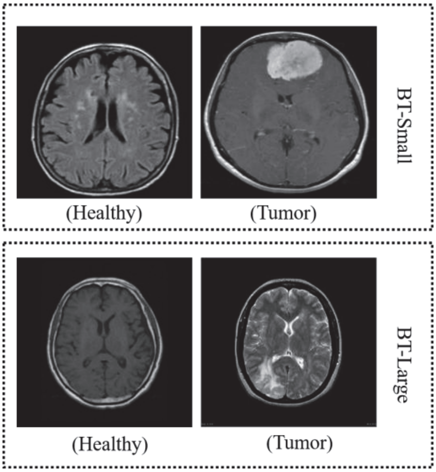


Figure 3. MRI samples from the employed datasets.

Table 1. Dataset Distribution.

Method	Dataset	Training Samples	Validation Samples
Study I	BT-small	177	76
Study II	BT-large	2100	900

4.2. Evaluation Parameters

Since the aim of detection frameworks is to predict unexpected data, successfully assessing the model's performance is crucial when developing automated systems. Thus, training and test sets evaluation illustrate the framework's generalization abilities. The most common method for assessing a classification model is performed via a confusion matrix (CM), a simple cross-tabulation of the actual and classified data for each class. Accuracy, F1-score, recall, and precision are a few classification metrics based on the CM used to assess the model's performance in this study. The F1-score is an effective tool for combining precision and recall in a single benchmark that contains features from both metrics. It is widely used in cases of data imbalance. The metrics used in this study, i.e., precision, recall, accuracy, and F1-score, are defined in Equations (11)–(14), respectively.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{11}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{12}$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{13}$$

$$\text{F1} - \text{Score} = (\text{TP})/(\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})) \tag{14}$$

TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively.

4.3. Results Obtained from the Proposed Framework

In this section, we highlight the results obtained from the proposed method on two datasets, i.e., BT-small (Study I) and BT-large (Study II). Table 2 compares the proposed method performance for individual FVs and EVs. The results show that the classifiers performed better on the novel FV than on single-model FVs. In Study I, KNN classified the novel FV with 96% accuracy, whereas SVM obtained 93.3% accuracy. On the other hand, in Study II, SVM achieved a 99% classification accuracy, whereas KNN obtained 98.7% accuracy. Figures 4 and 5 present precision, recall, and F1-score values obtained on EV for Study I and Study II, respectively.

Table 2. Proposed method results in terms of accuracy.

FV	Study I Accuracy %		Study II Accuracy %	
	SVM	**KNN**	**SVM**	**KNN**
VGG16	92.1	88.1	98.0	97.8
GLCM	72.0	84.0	96.1	96.0
GLCM + VGG16	93.3	96.0	99.0	98.7

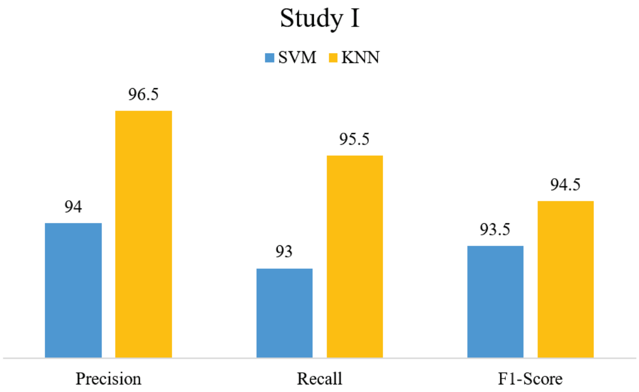


Figure 4. Study I Results in terms of PRE, REC, and F1-Score.

The classification performance of the proposed approach is shown in the CM in terms of actual and expected classes. The CM of the proposed FV obtained from Study I is presented in Figure 6. Here, (a) presents the CM obtained via KNN, while (b) shows the classification results for SVM. Study I obtains a classification accuracy of 96.0 using KNN and 93.3% using SVM, whereas, Figure 7 presents the CM of Study II. Part (a) denotes the CM obtained via KNN, and part (b) displays CM obtained via the SVM classifier. Study II obtains a classification accuracy of 98.7 using KNN and 99.0% using SVM. Study II achieved better performance due to a relatively larger database. These metrics demonstrate the reliability of our suggested strategy despite class imbalance because of a novel FV that contains discriminative and informative characteristics than an independent FV for the task of brain tumor classification.

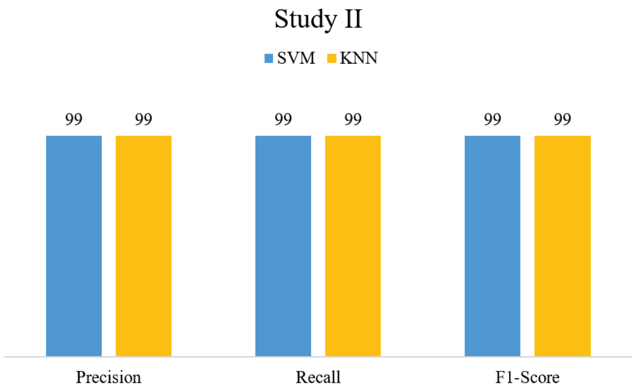


Figure 5. Proposed method results in terms of PRE, REC, and F1-Score.

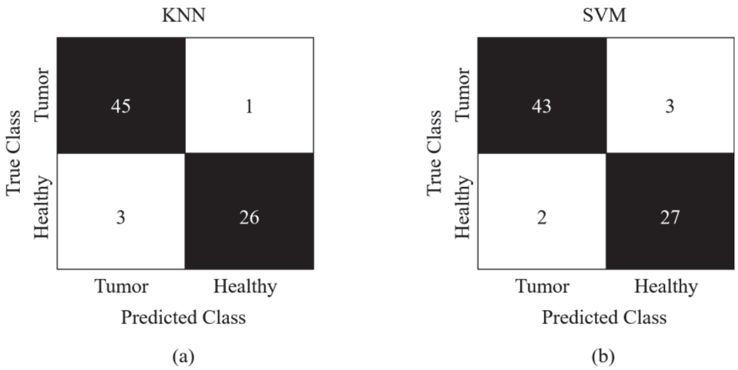


Figure 6. Confusion matrix obtained for Study I (**a**) KNN (**b**) SVM.

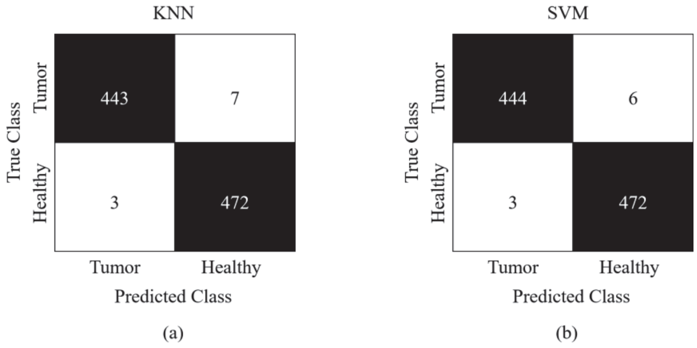


Figure 7. Confusion matrix obtained for Study II (**a**) KNN (**b**) SVM.

Figures 8 and 9 show Reciever Operating Curves (ROC) obtained from BT-Small and BT-Large databases, respectively. The original purpose of the ROC analysis was to examine radar signal noise during World War II [51]. Over the past few decades, ROC curves have gained prominence as a tool for evaluating the effectiveness of medical diagnostic systems. The curve depicts the trade-off between sensitivity and specificity. The model is more accurate when ROC is positioned at the top left corner. It is worth mentioning fluctuations do not affect that accuracy index due to arbitrarily defined thresholds [52].

The area determines the discriminative capacity of a framework under the curve (AUC), demonstrating how effectively it operates in a specific scenario. A robust model will have almost one AUC [53]. It can be seen that the proposed framework succeeded in achieving efficient results as the AUC nears one on the BT-Small database, whereas AUC = 1 in the case of the BT-Large database due to the presence of more images.

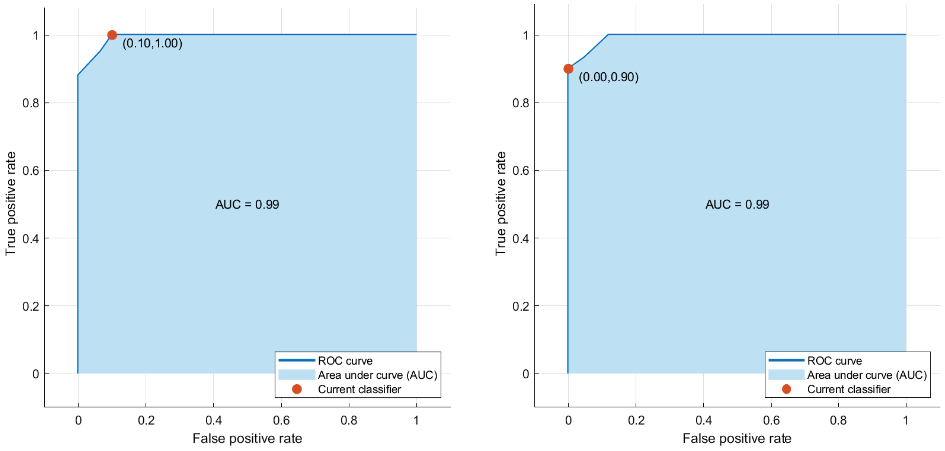


Figure 8. ROC Curve results obtained on BT-Small Dataset.

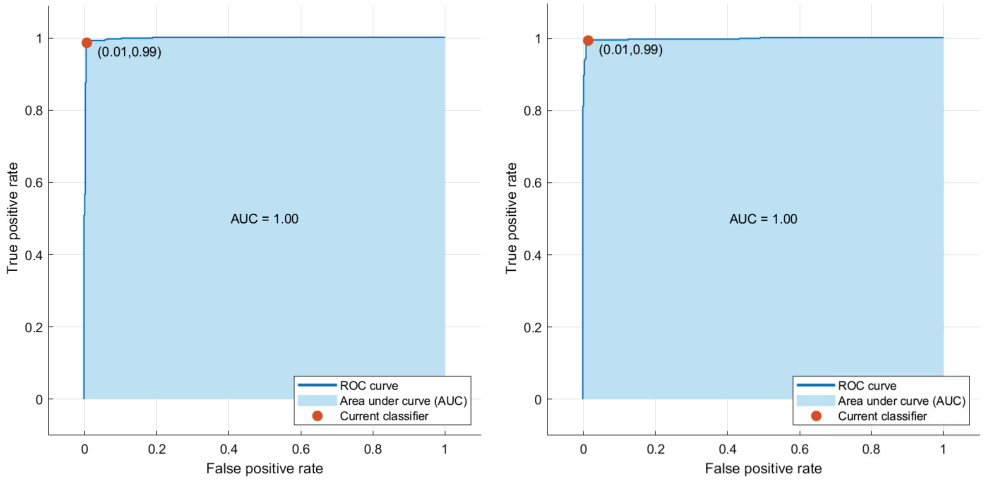


Figure 9. ROC Curve results obtained on BT-Large Dataset.

4.4. Cross-Dataset Validation

We investigated the cross-dataset detection capability of our suggested technique. The primary goal of testing the model on cross-dataset scenarios is to assess the generalization potential of the suggested technique. We used BT-Small and BT-large datasets to train our approach, and we tested Study I's performance using Study II's dataset and vice versa. Detailed results of cross-dataset validation in terms of accuracy are presented in Table 3. Technique in Study I obtained 92% accuracy via SVM and 90.0% accuracy via KNN on the MRI images acquired from the BT-Large dataset. At the same time, Study II obtained 99.2% accuracy via KNN and 99.6% accuracy via SVM on MRI images from the BT-Small dataset. The classifiers in Study II performed better than those in Study I due to a larger MRI dataset. The results also show that our suggested methods can detect brain cancer in unseen MRI samples. This leads to the conclusion that the suggested strategy is reliable and effective for locating and classifying brain malignancies from MRI data.

Table 3. Cross Dataset Validation.

Classifiers/Method	**Accuracy %**	
	Study I	**Study II**
SVM	92.0	99.6
KNN	90.0	99.2

4.5. Comparison with Existing Approaches

Recent medical image processing software developments have made it easier for medical professionals to identify ailments early on. These developments are helping them in some medical specialties, including disease diagnosis, as well as decision-making for clinical applications. Healthcare centers generate tons of medical records daily. Doctors and scientists looking for the best ways to utilize these ever-growing volumes of data effectively are helped by medical informatics research [54,55]. Early discovery and adequate treatment choices are critical for effectively treating brain tumor disorders. These treatment options are based on the type of tumor and its stage during detection time. The traditional identification methods use simple ML-oriented algorithms that only extract a few features [56]. This study introduces a unique feature ensemble-based method for precisely classifying brain cancers from MR images.

This paper presents a unique FV-based ensemble of hand-engineered and deep features for brain tumor classification. We investigated the usefulness of the proposed model by comparing our approach to existing brain tumor classification techniques in Table 4. Irsheidat et al. [57] proposed a method for detecting brain tumors using artificial CNNs. They trained a deep CNN on a dataset of MRI images to differentiate between normal and tumor images. The proposed method employed data augmentation techniques to increase the training dataset's size and improve the model's generalization. The authors reported an accuracy of 96.9% in detecting brain tumors using their proposed deep CNN model. However, the system attained lower accuracy than the proposed method.

Table 4. Comparison with existing techniques.

Reference	**Technique**	**ACC(%)**
Irsheidat et al. [32]	CNN	96.7
Kesav et al. [33]	Two Channel CNN	98
Tazin et al. [57]	Mobile Net V2	92
Sharma et al. [56]	VGG-19	94.7
Masood et al. [21]	Mask RCNN	98.3
Proposed Study I	**VGG16 + GLCM**	**96.0**
Proposed Study II		**99.0**

On the contrary, Mask RCNN was employed by Masood et al. [21] to detect brain malignancies in MRI images. They used bounding boxes to pinpoint the tumor site in the first stage and then classified the tumors. Due to the need for tumor segmentation before the classification phase, this technique is computationally intensive. Kesav et al. [33] proposed an architecture for detecting and classifying brain tumors using a combination of region-based CNN and two-channel CNN. The proposed architecture used RCNN to identify regions of interest in MRIs and then fed these regions into a two-channel CNN for classification. The two-channel CNN comprises two separate CNN models, one for the segmentation of the tumor region and the other for the classification of the segmented tumor. However, the system is computationally expensive. Sharma et al. [56] used the concept of transfer learning for brain tumor detection. They employed VGG-16 architecture that attained 94.&% accuracy on the Kaggle database. Compared to previous methods, our suggested method is efficient, using a hybrid feature set consisting of deep features and GLCM features to identify and categorize brain tumors. Moreover, the proposed framework does not require segmentation before categorization. The ensemble of FVs obtained from deep and statistical methods results in a more discriminative feature representation than a single model FV. The proposed EV achieved the highest accuracy of 96% in Study I and 99% in Study II. From the results, it is evident that the proposed structure is robust and provides better classification results compared to existing studies.

5. Conclusions

This paper proposes a novel brain tumor detection and classification method using ensemble learning that integrates hand-crafted and deep features in a single FV. The hand-crafted are computed via GLCM, whereas the deep features are extracted from VGG-16. Both the independent FVs are then serially combined, resulting in a single FV (called EV) that is more discriminate and informative because of the concatenation of textural and deep features. EV is then supplied to well-known classification frameworks such as SVM and KNN. Our method is individually trained and validated on two benchmark databases and obtained maximum accuracy of 96% on the BT-Small dataset and 99% on the BT-Large database. We also validated the performance of our model via cross-dataset validation and compared the proposed technique with the existing systems. The results show the robustness of the proposed method and can be deployed in the real environment to detect brain tumors from MRI images accurately. In the future, we will gather medical images of other modalities and employ other CNN architectures to make the system more efficient. Furthermore, we are willing to test the proposed approach to perform the distinctive category-wise distribution of brain MRI images to determine the specific type of tumors as well.

Author Contributions: Software, H.K.; Validation, R.A., J.K. and R.G.; Formal analysis, H.K. and R.A.; Investigation, M.N.; Resources, J.K.; Data curation, J.K. and R.G.; Writing—original draft, H.K.; Writing—review & editing, R.G.; Visualization, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization Support Program (IITP-2023-RS-2022-00156287) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Anitha, R.; Raja, D.S.S. Development of computer-aided approach for brain tumor detection using random forest classifier. *Int. J. Imaging Syst. Technol.* **2017**, *28*, 48–53. [CrossRef]
2. Lu, J.; Nguyen, M.; Yan, W.Q. Deep learning methods for human behavior recognition. In Proceedings of the 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ), Wellington, New Zealand, 25–27 November 2020; pp. 1–6.
3. Al-Okaili, R.N.; Krejza, J.; Wang, S.; Woo, J.H.; Melhem, E.R. Advanced MR Imaging Techniques in the Diagnosis of Intraaxial Brain Tumors in Adults. *Radiographics* **2006**, *26*, S173–S189. [CrossRef] [PubMed]
4. Badža, M.M.; Barjaktarović, M. Segmentation of Brain Tumors from MRI Images Using Convolutional Autoencoder. *Appl. Sci.* **2021**, *11*, 4317. [CrossRef]

5. Nisar, D.-E.; Amin, R.; Shah, N.-U.; Al Ghamdi, M.A.; Almotiri, S.H.; Alruily, M. Healthcare Techniques Through Deep Learning: Issues, Challenges and Opportunities. *IEEE Access* **2021**, *9*, 98523–98541. [CrossRef]
6. Swati, Z.N.K.; Zhao, Q.; Kabir, M.; Ali, F.; Ali, Z.; Ahmed, S.; Lu, J. Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput. Med. Imaging Graph.* **2019**, *75*, 34–46. [CrossRef]
7. Anaraki, A.K.; Ayati, M.; Kazemi, F. Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybern. Biomed. Eng.* **2018**, *39*, 63–74. [CrossRef]
8. Habib, H.; Amin, R.; Ahmed, B.; Hannan, A. Hybrid algorithms for brain tumor segmentation, classification and feature extraction. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 1–22.
9. Hussain, S.; Anwar, S.M.; Majid, M. Segmentation of glioma tumors in brain using deep convolutional neural network. *Neurocomputing* **2018**, *282*, 248–261. [CrossRef]
10. Muhammad, K.; Khan, S.; Del Ser, J.; de Albuquerque, V.H.C. Deep Learning for Multigrade Brain Tumor Classification in Smart Healthcare Systems: A Prospective Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 507–522. [CrossRef]
11. Mzoughi, H.; Njeh, I.; Wali, A.; Ben Slima, M.; BenHamida, A.; Mhiri, C.; Ben Mahfoudhe, K. Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification. *J. Digit. Imaging* **2020**, *33*, 903–915. [CrossRef]
12. Sejuti, Z.A.; Islam, M.S. An Efficient Method to Classify Brain Tumor using CNN and SVM. In Proceedings of the IEEE 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 5–7 January 2021; pp. 644–648.
13. Abiwinanda, N.; Hanif, M.; Hesaputra, S.T.; Handayani, A.; Mengko, T.R. Brain tumor classification using convolutional neural network. In Proceedings of the World Congress on Medical Physics and Biomedical Engineering 2018, Prague, Czech Republic, 3–5 May 2019; pp. 183–189.
14. Khan, A.H.; Abbas, S.; Khan, M.A.; Farooq, U.; Khan, W.A.; Siddiqui, S.Y.; Ahmad, A. Intelligent Model for Brain Tumor Identification Using Deep Learning. *Appl. Comput. Intell. Soft Comput.* **2022**, *2022*, 1–10. [CrossRef]
15. Alanazi, M.F.; Ali, M.U.; Hussain, S.J.; Zafar, A.; Mohatram, M.; Irfan, M.; AlRuwaili, R.; Alruwaili, M.; Ali, N.H.; Albarrak, A.M. Brain Tumor/Mass Classification Framework Using Magnetic-Resonance-Imaging-Based Isolated and Developed Transfer Deep-Learning Model. *Sensors* **2022**, *22*, 372. [CrossRef] [PubMed]
16. Afshar, P.; Plataniotis, K.N.; Mohammadi, A. Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 1368–1372.
17. Noreen, N.; Palaniappan, S.; Qayyum, A.; Ahmad, I.; Alassafi, M.O. Brain Tumor Classification Based on Fine-Tuned Models and the Ensemble Method. *Comput. Mater. Contin.* **2021**, *67*, 3967–3982. [CrossRef]
18. Kang, J.; Ullah, Z.; Gwak, J. MRI-Based Brain Tumor Classification Using Ensemble of Deep Features and Machine Learning Classifiers. *Sensors* **2021**, *21*, 2222. [CrossRef]
19. Waghmare, V.K.; Kolekar, M.H. Brain Tumor Classification Using Deep Learning. In *Internet of Things for Healthcare Technologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 155–175.
20. Naser, M.A.; Deen, M.J. Brain tumor segmentation and grading of lower-grade glioma using deep learning in MRI images. *Comput. Biol. Med.* **2020**, *121*, 103758. [CrossRef]
21. Masood, M.; Nazir, T.; Nawaz, M.; Mehmood, A.; Rashid, J.; Kwon, H.-Y.; Mahmood, T.; Hussain, A. A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images. *Diagnostics* **2021**, *11*, 744. [CrossRef]
22. Sajjad, M.; Khan, S.; Muhammad, K.; Wu, W.; Ullah, A.; Baik, S.W. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *J. Comput. Sci.* **2018**, *30*, 174–182. [CrossRef]
23. Amin, J.; Sharif, M.; Yasmin, M.; Fernandes, S.L. A distinctive approach in brain tumor detection and classification using MRI. *Pattern Recognit. Lett.* **2017**, *139*, 118–127. [CrossRef]
24. Kaplan, K.; Kaya, Y.; Kuncan, M.; Ertunç, H.M. Brain tumor classification using modified local binary patterns (LBP) feature extraction methods. *Med. Hypotheses* **2020**, *139*, 109696. [CrossRef]
25. Bahadure, N.B.; Ray, A.K.; Thethi, H.P. Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. *Int. J. Biomed. Imaging* **2017**, *2017*, 1–12. [CrossRef]
26. Garg, G.; Garg, R. Brain Tumor Detection and Classification based on Hybrid Ensemble Classifier. *arXiv* **2021**, arXiv:2101.00216.
27. Minz, A.; Mahobiya, C. MR image classification using adaboost for brain tumor type. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 5–7 January 2017; pp. 701–705.
28. Raja, P.S.J.B.; Engineering, B. Brain tumor classification using a hybrid deep autoencoder with Bayesian fuzzy clustering-based segmentation approach. *Biocybern. Biomed. Eng.* **2020**, *40*, 440–453. [CrossRef]
29. Ali, F.; Khan, S.; Abbas, A.W.; Shah, B.; Hussain, T.; Song, D.; Ei-Sappagh, S.; Singh, J. A Two-Tier Framework Based on GoogLeNet and YOLOv3 Models for Tumor Detection in MRI. *Comput. Mater. Contin.* **2022**, *72*, 73–92. [CrossRef]
30. Das, S.; Nayak, G.; Saba, L.; Kalra, M.; Suri, J.S.; Saxena, S. An artificial intelligence framework and its bias for brain tumor segmentation: A narrative review. *Comput. Biol. Med.* **2022**, *143*, 105273. [CrossRef]
31. Khan, M.A.; Ashraf, I.; Alhaisoni, M.; Damaševičius, R.; Scherer, R.; Rehman, A.; Bukhari, S.A.C. Multimodal Brain Tumor Classification Using Deep Learning and Robust Feature Selection: A Machine Learning Application for Radiologists. *Diagnostics* **2020**, *10*, 565. [CrossRef]

32. Irsheidat, S.; Duwairi, R. Brain tumor detection using artificial convolutional neural networks. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 197–203.
33. Kesav, N.; Jibukumar, M. Efficient and low complex architecture for detection and classification of Brain Tumor using RCNN with Two Channel CNN. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *34*, 6229–6242. [CrossRef]
34. Shahajad, M.; Gambhir, D.; Gandhi, R. Features extraction for classification of brain tumor MRI images using support vector machine. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 767–772. [CrossRef]
35. Mathswork. Feature Extraction for Machine Learning and Deep Learning. Available online: https://www.mathworks.com/discovery/feature-extraction.html (accessed on 17 August 2022).
36. Fu, P.; Chu, L.; Hou, Z.; Xing, J.; Gao, J.; Guo, C. Deep learning based velocity prediction with consideration of road structure. In Proceedings of the 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), Tianjin, China, 29–31 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
37. Buduma, N.; Buduma, N.; Papa, J. *Fundamentals of Deep Learning*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2022.
38. Sedik, A.; Hammad, M.; El-Samie, F.E.A.; Gupta, B.B.; El-Latif, A.A.A. Efficient deep learning approach for augmented detection of Coronavirus disease. *Neural Comput. Appl.* **2021**, *34*, 11423–11440. [CrossRef] [PubMed]
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
40. Nadeem, M.W.; Ghamdi, M.A.A.; Hussain, M.; Khan, M.A.; Khan, K.M.; Almotiri, S.H.; Butt, S.A. Brain tumor analysis empowered with deep learning: A review, taxonomy, and future challenges. *Brain Sci.* **2020**, *10*, 118. [CrossRef]
41. Kumar, S.; Fred, A.L.; Padmanabhan, P.; Gulyas, B.; Kumar, H.A.; Miriam, L.J. Deep Learning Algorithms in Medical Image Processing for Cancer Diagnosis: Overview, Challenges and Future. *Deep. Learn. Cancer Diagn.* **2021**, 37–66.
42. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
43. Nanni, L.; Ghidoni, S.; Brahnam, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognit.* **2017**, *71*, 158–172. [CrossRef]
44. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *3*, 610–621. [CrossRef]
45. Humeau-Heurtier, A. Texture Feature Extraction Methods: A Survey. *IEEE Access* **2019**, *7*, 8975–9000. [CrossRef]
46. Suthaharan, S. Support vector machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 207–235.
47. Kibriya, H.; Amin, R.; Alshehri, A.H.; Masood, M.; Alshamrani, S.S.; Alshehri, A. A Novel and Effective Brain Tumor Classification Model Using Deep Feature Fusion and Famous Machine Learning Classifiers. *Comput. Intell. Neurosci.* **2022**, *2022*, 7897669. [CrossRef]
48. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
49. Cai, J.; Li, J.; Li, W.; Wang, J. Deeplearning model used in text classification. In Proceedings of the 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 14–16 December 2018; pp. 123–126.
50. Semberecki, P.; Maciejewski, H. Deep learning methods for subject text classification of articles. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3–6 September 2017; pp. 357–360.
51. Zou, K.H.; O'Malley, A.J.; Mauri, L.J.C. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation* **2007**, *115*, 654–657. [CrossRef]
52. Hajian-Tilaki, K. Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian J. Intern. Med.* **2013**, *4*, 627.
53. Ekelund, S. Roc Curves—What are they and how are they used? *Point Care* **2012**, *11*, 16–21. [CrossRef]
54. Jiang, F.; Jiang, Y.; Zhi, H.; Dong, Y.; Li, H.; Ma, S.; Wang, Y.; Dong, Q.; Shen, H.; Wang, Y. Artificial intelligence in healthcare: Past, present and future. *Stroke Vasc. Neurol.* **2017**, *2*, 230–243. [CrossRef]
55. Sultan, H.H.; Salem, N.M.; Al-Atabany, W. Multi-Classification of Brain Tumor Images Using Deep Neural Network. *IEEE Access* **2019**, *7*, 69215–69225. [CrossRef]
56. Sharma, S.; Gupta, S.; Gupta, D.; Juneja, A.; Khatter, H.; Malik, S.; Bitsue, Z.K. Deep Learning Model for Automatic Classification and Prediction of Brain Tumor. *J. Sens.* **2022**, *2022*, 1–11. [CrossRef]
57. Tazin, T.; Sarker, S.; Gupta, P.; Ibn Ayaz, F.; Islam, S.; Khan, M.M.; Bourouis, S.; Idris, S.A.; Alshazly, H. A Robust and Novel Approach for Brain Tumor Classification Using Convolutional Neural Network. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–11. [CrossRef] [PubMed]

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

Deep Learning Framework for Liver Segmentation from T_1-Weighted MRI Images

Md. Sakib Abrar Hossain [1,2], Sidra Gul [3,4], Muhammad E. H. Chowdhury [2,*], Muhammad Salman Khan [2], Md. Shaheenur Islam Sumon [2], Enamul Haque Bhuiyan [5], Amith Khandakar [2], Maqsud Hossain [1], Abdus Sadique [1], Israa Al-Hashimi [6], Mohamed Arselene Ayari [7], Sakib Mahmud [2] and Abdulrahman Alqahtani [8,9]

[1] NSU Genome Research Institute (NGRI), North South University, Dhaka 1229, Bangladesh
[2] Department of Electrical Engineering, Qatar University, Doha 2713, Qatar
[3] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Peshawar 25000, Pakistan
[4] Artificial Intelligence in Healthcare, IIPL, National Center of Artificial Intelligence, Peshawar 25000, Pakistan
[5] Center for Magnetic Resonance Research, University of Illinois Chicago, Chicago, IL 60607, USA
[6] Hamad Medical Corporation, Doha 3050, Qatar
[7] Department of Civil Engineering, Qatar University, Doha 2713, Qatar
[8] Department of Medical Equipment Technology, College of Applied, Medical Science, Majmaah University, Majmaah City 11952, Saudi Arabia
[9] Department of Biomedical Technology, College of Applied Medical Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
* Correspondence: mchowdhury@qu.edu.qa

Abstract: The human liver exhibits variable characteristics and anatomical information, which is often ambiguous in radiological images. Machine learning can be of great assistance in automatically segmenting the liver in radiological images, which can be further processed for computer-aided diagnosis. Magnetic resonance imaging (MRI) is preferred by clinicians for liver pathology diagnosis over volumetric abdominal computerized tomography (CT) scans, due to their superior representation of soft tissues. The convenience of Hounsfield unit (HoU) based preprocessing in CT scans is not available in MRI, making automatic segmentation challenging for MR images. This study investigates multiple state-of-the-art segmentation networks for liver segmentation from volumetric MRI images. Here, T1-weighted (in-phase) scans are investigated using expert-labeled liver masks from a public dataset of 20 patients (647 MR slices) from the Combined Healthy Abdominal Organ Segmentation grant challenge (CHAOS). The reason for using T1-weighted images is that it demonstrates brighter fat content, thus providing enhanced images for the segmentation task. Twenty-four different state-of-the-art segmentation networks with varying depths of dense, residual, and inception encoder and decoder backbones were investigated for the task. A novel cascaded network is proposed to segment axial liver slices. The proposed framework outperforms existing approaches reported in the literature for the liver segmentation task (on the same test set) with a dice similarity coefficient (DSC) score and intersect over union (IoU) of 95.15% and 92.10%, respectively.

Keywords: deep learning; automated liver segmentation; MRI; diagnostic radiology; T_1-weighted contrast

Citation: Hossain, M.S.A.; Gul, S.; Chowdhury, M.E.H.; Khan, M.S.; Sumon, M.S.I.; Bhuiyan, E.H.; Khandakar, A.; Hossain, M.; Sadique, A.; Al-Hashimi, I.; et al. Deep Learning Framework for Liver Segmentation from T_1-Weighted MRI Images. *Sensors* **2023**, *23*, 8890. https://doi.org/10.3390/s23218890

Academic Editors: Dae-Ki Kang and Sukho Lee

Received: 23 June 2023
Revised: 8 August 2023
Accepted: 15 August 2023
Published: 1 November 2023

1. Introduction

Over the past decade, remarkable advancements in deep learning (DL) algorithms have led to a rapid transformation in the field of radiology. DL-aided diagnostics have achieved exceptional accuracy in detecting abnormalities in various domains such as ophthalmology, respiratory, and breast imaging. In some cases, multimodal DL solutions now exhibit accuracy levels comparable to expert radiologists. The high performance and clinically satisfactory outcomes achieved through computer-aided diagnostic radiology were previously considered inconceivable [1–3].

Semantic segmentation is a prerequisite for any DL-driven diagnostic task, as it allows the model to learn from the region of interest. Formerly, for semantic segmentation tasks in radiology/medical imaging, distinct mathematical models were implemented, but such approaches often lacked a generalized solution. Deep learning-based segmentation tasks outperform conventional mathematical modeling-based approaches. Segmentation has always helped in improving the performance of computer-aided diagnosis [4,5]. Q. Dou et al. present a unique 3D deeply supervised network (3D DSN) explicitly designed for liver segmentation from CT data [6]. The network incorporates deep supervision to enhance optimization and discrimination capabilities during the learning process, resulting in competitive segmentation results compared to state-of-the-art approaches, along with improved processing speeds. In another study, C. Chen et al. propose an innovative method for lung lesion segmentation in CT scans of COVID-19 patients. Their approach involves region-of-interest extraction and employs a 3D network with attention mechanisms to enhance segmentation accuracy [7]. Additionally, C. Chen et al. introduce a rapid and precise lung segmentation technique, utilizing the edge-weighted random walker algorithm with spatial and clustering information to achieve a heightened accuracy and reduced segmentation time [8]. Similarly, P. Hu et al. develop a liver segmentation framework by integrating a 3D convolutional neural network (CNN) with globally optimized surface evolution. Their approach demonstrates effective segmentation outcomes suitable for clinical applications [9]. Together, these contributions significantly enhance the field of automated organ segmentation, offering valuable insights for medical imaging research and clinical implementations.

However, such a segmentation task in an anatomical paradigm, i.e., the identification and delineation of an anatomical area or structure in magnetic resonance imaging (MRI), encounters a colossal amount of complexity. The complexity can be due to topology, spatial distance, location, relative motion, texture, geometrical structure, and other varying anatomical information. As a consequence, anatomical segmentation has always been a demanding task. In particular, compared to other anatomical structures, very few significant works can be found that focus on liver segmentation [10–12].

For any deep learning-based liver disease diagnosis system, precise automated liver segmentation is indispensable. However, similar to any anatomical segmentation task, it is extensively challenging. This is due to the fact that, compared with other abdominal organs, its anatomy can noticeably differ with patients and clinical conditions. Additionally, the liver's proximity to contiguous abdominal organs (the spleen and kidneys) generates substantial ambiguity [13,14].

However, recent research has demonstrated excellent results for deep neural network (DNN)-based liver segmentation tasks from volumetric abdominal computed tomography (CT) images. Tang et al. [15] achieved a dice similarity coefficient (DSC) of 98% in the liver segmentation task from a plain CT scan using a modified multiscaled convolutional neural network (CNN). Hu et al. [9] used a three-dimensional CNN for the same task and achieved a high performance of around a 97.25% dice similarity coefficient. These works utilized Hounsfield unit (HoU) scaling as a hyperparameter for image enhancement in the preprocessing stage [16]. The review by Xiang et al. [17] observed that, in terms of liver segmentation from magnetic resonance imaging (MRI) scans, high performance could not be achieved and also very little significant work exists in this domain. Owing to the absence of such homogeneous HoU-based image enhancement convenience, in terms of automated liver segmentation from volumetric abdominal MR scans, achieving a similarly high performance to CT images is challenging.

Moreover, MRI scans are extensively adopted by clinicians for liver pathology investigation, due to their superior contrast and spatial resolution for soft tissues compared to CT scans [18,19]. CT scans can provide solid anatomical information. On the contrary, MRI demonstrates high signal intensity in comparison to CT scans. As a result, both anatomical and physiological information can be derived from MRI scans. In particular, both CT and MRI can provide accurate anatomical information about the liver lesion or haemangioma; however, MRI scans can provide a further important basis for screening benign or malignant types [20–22].

The above studies demonstrate the significance of liver segmentation, specifically from volumetric MRI scans, as this modality is favored by clinicians in relation to pathological diagnosis. In this regard, liver segmentation from MRI scans holds significant importance. A. Mostafa et al. investigated a whale optimization algorithm for liver segmentation from MRI scans [23]. A. Hänsch et al. studied multimodal training and three-dimensional CNN for the task [24]. X. Zhong et al. used deep action learning with a 3D UNet [25], and P. Pandey et al. investigated contrastive semisupervised learning for the liver segmentation task [26] on the CHAOS abdominal MRI dataset [27]. D. Mitta et al. implemented a weighted UNet with attention gates for the liver segmentation task [28] on the same dataset, and J. Hong et al. achieved a slightly better performance using a source-free unsupervised UNet [29]. X. Wang et al. investigated a bidirectional search of the neural net for the task [30]. Additionally, S. Mulay et al. used a geometric edge enhancement-based mask R-CNN [31]. The more recent work of L. Zbinden et al. achieved better performance than previous research for liver segmentation on the same testing set by implementing nnUNet on T_1-weighted MRI slices [32].

In this research, we investigated 24 state-of-the-art segmentation networks for liver segmentation tasks from T_1-weighted MR scans using a publicly available dataset, which annotated ground truths for the liver segmentation of 20 patients. The prospect of predicting a precise mask from T_1-weighted MR scans is higher as fat (and protein) contents are brighter and more distinguishable in such a group. The investigation explores state-of-the-art segmentation networks, such as UNet, UNet++, and feature pyramid network (FPN) segmentation networks with varying dense encoder backbones, along with various image enhancement techniques in the preprocessing stage. The proposed cascaded network showed superior performance to many high-performance state-of-the-art approaches on the same test set. Finally, we developed a software prototype by deploying our proposed DL model in a cloud server for public usage. The cross-platform software is open source and can be accessed from http://130.211.209.103/projects/the-big-mri-project-beta, accessed on 9 August 2023.

The main contributions of the research are listed below.

- This research extensively investigates state-of-the-art approaches for precise liver segmentation from T1-weighted abdominal MR scans to facilitate clinicians with AI-driven assistance for liver pathology diagnosis;
- This research investigates the effects of multiple image enhancement techniques for automated liver segmentation tasks from MR scans;
- This research proposes a novel cascaded network for the liver segmentation task that demonstrated state-of-the-art performance compared to the literature;
- The proposed model was deployed in a cloud server for demonstration purposes so that clinicians can directly benefit from the results of this investigation.

2. Materials and Methods

The brief methodology of the research is explained in Figure 1. The methodology will be discussed in detail in the following section.

2.1. Dataset

The dataset was collected from the Combined Healthy Abdominal Organ Segmentation (CHAOS) grant challenge [27]. The public portion of the CHAOS dataset includes computed tomography (CT) and magnetic resonance imaging (MRI) abdominal scans of 20 patients, in the Digital Imaging and Communications in Medicine (DICOM) format. The ground truths (GT) were provided from the source, which includes masks for the right kidney, left kidney, liver, and spleen. The ground truth masks were annotated by certified radiologists. All scans are of healthy patients. The MRI scans include T_1-weighted in-phase and out-phase, along with T_2-weighted scans, which are discussed in the next subsection.

Each T_1 scan includes 26 to 56 slices; for 20 patients the total number of T_1-weighted slices is 647 [27].

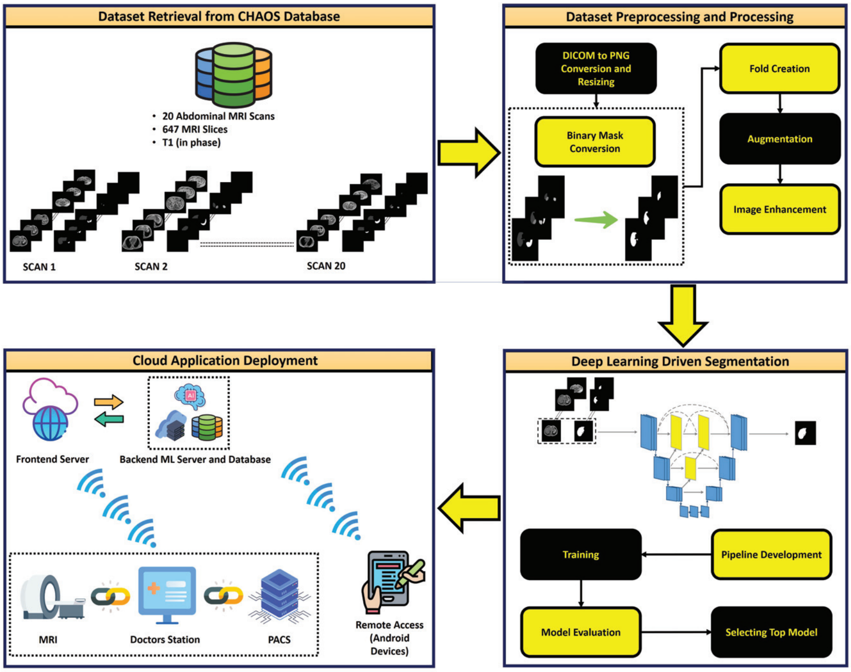


Figure 1. Flow diagram explaining methodology for automated liver segmentation from T_1-weighted MRI scans.

2.2. Selecting Task-Specific Contrast Group

Among different contrast-enhanced groups (in-phase T_1-weighted, out-phase T_1-weighted, and T_2-weighted), specific groups were chosen by analyzing the relevant abdominal anatomy and attributes of the available contrast-enhanced groups.

2.2.1. Relevant Abdominal Anatomy

The supplied masks (left kidney, right kidney, liver, and spleen) lie in close proximity to each other in the abdominal region, leading to a colossal amount of ambiguity in distinguishing any of the organs. The anatomy of these organs is briefly visualized in Figure 2 [33]. The superior part of the liver (left lobe) lies within the epigastric and left hypochondriac regions. It is in close proximity to the spleen and rests in front of the spleen in terms of the axial plane. The middle part of the liver resides above the umbilical region. The inferior part of the liver is just in front of the upper pole of the right kidney, which occupies the right lumbar region. The left kidney lies in the left lumbar region just below the spleen. Therefore, such close proximity generates an enormous amount of complexity and obscurity in automated abdominal organ segmentation tasks using machine learning.

2.2.2. T_1- and T_2-Weighted Images

The T_1 and T_2 parameter represents relaxation time for longitudinal (M_z) and transverse (M_t) magnetization components for each proton. T_1 is noted as the spin–spin relaxation phenomenon, and T_2 is noted as the spin–lattice relaxation phenomenon. When the macroscopic magnetizing vector for each voxel is M_o, then the relationship among the magnetization components and T_2, T_1 is denoted as [34]

$$M_t(t) = M_o \sin \alpha e^{\frac{-t}{T_2}} \tag{1}$$

$$M_z(t) = M_o \cos \alpha e^{\frac{-t}{T_1}} + M_o(1 - e^{\frac{-t}{T_1}}) \tag{2}$$

where α denotes the flip angle, which represents a rotation in net magnetization. Characteristically, the T_1 tissue relaxation time is always larger than T_2. The relaxation times vary broadly with tissue attributes and characteristics. These varying intervals can also be used to distinguish between healthy and abnormal tissues. Table 1 denotes T_1 and T_2 values for relevant abdominal tissues [35].

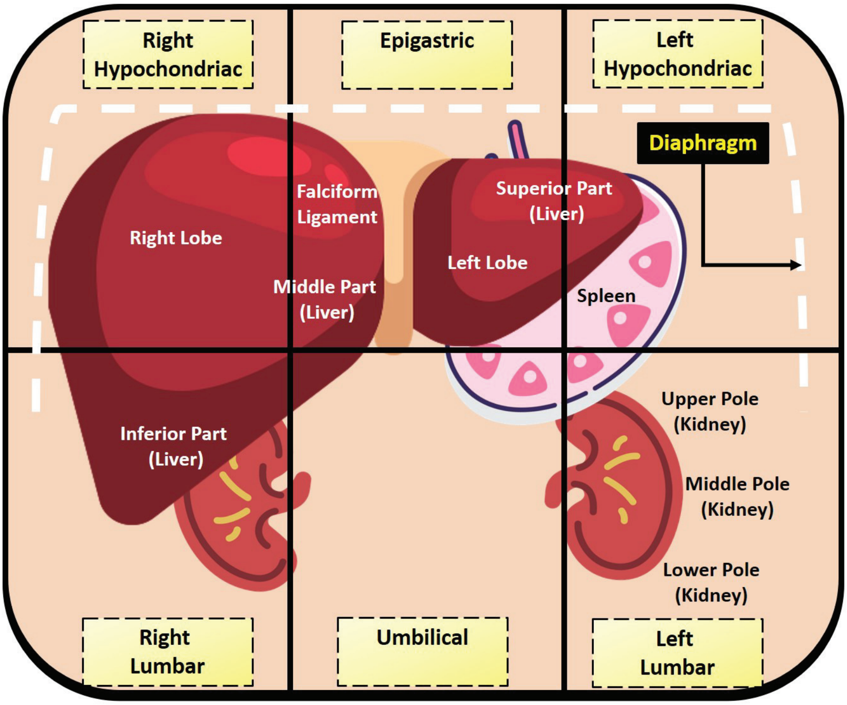


Figure 2. Superficial visualization of relevant abdominal anatomy for describing underlying ambiguity in the segmentation task.

Table 1. Average T_1 and T_2 relaxation time (msec) for 1.5 T and 3.0 T MRI scans.

Tissue	**1.5 T**		**3.0 T**	
	T_1 **(msec)**	T_2 **(msec)**	T_1 **(msec)**	T_2 **(msec)**
Kidney	966–1412	85–87	1142–1545	76–81
Liver	586	46	809	34
Spleen	1057	79	1328	79
Lipid	343	58	382	68

The relation among the image intensity of each voxel $I(x,y)$, the tissue density $\rho(x,y)$, the echo time (TE), and the repetition time (TR) can be denoted as

$$I(x,y) = \rho(x,y)\frac{(1-e^{\frac{TR}{T_1}})\sin\alpha}{1-e^{\frac{TR}{T_1}}\cos\alpha}e^{\frac{TE}{T_2}} \tag{3}$$

In Equation (3), α is optimized by following

$$\alpha_{Ernst} = \cos^{-1} e^{\frac{TR}{T_1}} \tag{4}$$

when $TE \ll T_2$, and either $\alpha \sim \alpha_{Ernst}$ or $TR \sim T_1$, then the image is defined as T_1-weighted. Moreover, the image is defined as T_2-weighted when $TE > T_2$, and either $\alpha \ll \alpha_{Ernst}$ or $TR \gg T_1$ [36].

In accordance with its definitions, fat (and protein) content in T_1-weighted MRI scans is brighter. Owing to such characteristics, the liver is more distinguishable in T_1-weighted MRI scans. Figure 3 shows sample T_1- and T_2-weighted slices of different axial views. It is clear from the figure that for T_1-weighted in-phase scans, the liver is far more distinguishable (even in the slices where the liver is small) in the inferior part of the liver, and the superior part of the liver in the axial view. In the MRI slices where the liver is larger (i.e., the middle part of the liver), both in-phase and out-phase T_1-weighted scans can be used. As T_1-weighted out-phase scans represent out-of-phase protons, a darker boundary can be noticed around regions of varying intensities. As a result, unwanted artifacts are introduced in these slices.

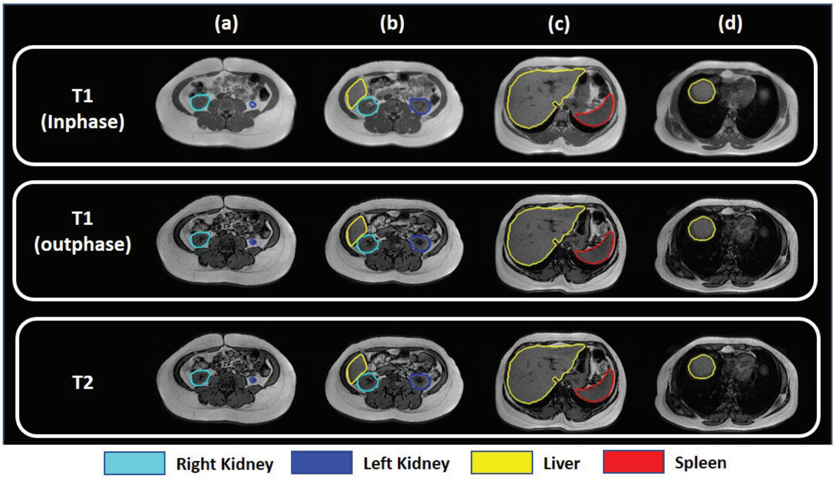


Figure 3. Visualization of MRI slices from (**a**) upper pole of the kidney, (**b**) inferior part of the liver, (**c**) middle part of the liver, and (**d**) superior part of the liver for different types of data available in the dataset.

Due to such attributes among different contrast-enhanced MRI scans, in-phase T_1-weighted contrast-enhanced scans were selected for the liver segmentation task. Such groups provide initially enhanced images, which can contribute to boosting the performance of deep neural networks.

2.3. Dataset Preprocessing

Firstly, in-phase T_1-weighted 647 DICOM slices were converted to PNG format in order to optimize the preprocessing and processing steps. In the ground truth mask, there are multiple organs (right kidney, left kidney, liver, and spleen) present. Binary masks are generated for the liver alone. Each slice and GT mask pair is then resized to 256 × 256 dimensions from their original 512 × 512 dimensions. Reducing the size of the dataset offers notable benefits in terms of enhancing computational efficiency during the training process of segmentation networks.

In order to ensure the data are ready for the machine learning investigation, there are important steps that include fold creation from the preprocessed dataset. Fold creation invovles dividing the data into the training set, validation set, and testing set for five folds. In order to avoid biases during training, it is important to make sure that the dataset is balanced; this is achieved by the augmentation of the training set. Finally, the authors investigated different image enhancement techniques for each of the created folds. Figure 4 represents techniques for fold creation and augmentation, which performed following the literature in [37–39]. Image enhancement techniques are demonstrated in Figure 5.

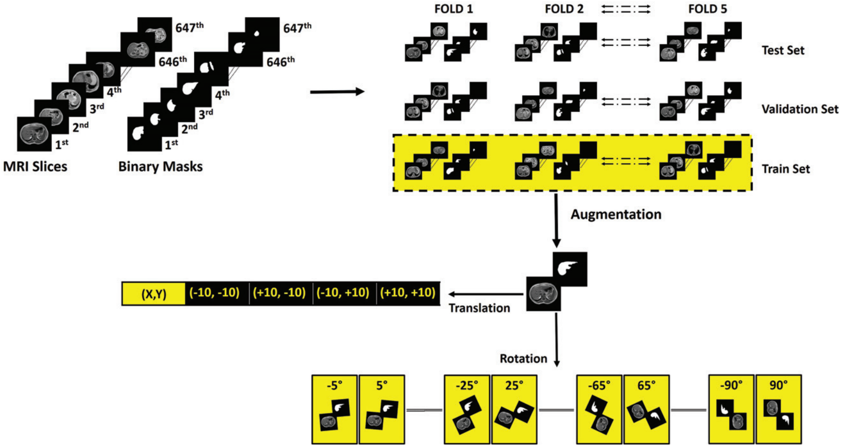


Figure 4. Flow diagram explaining the methodology for fold creation and augmentation in the training set.

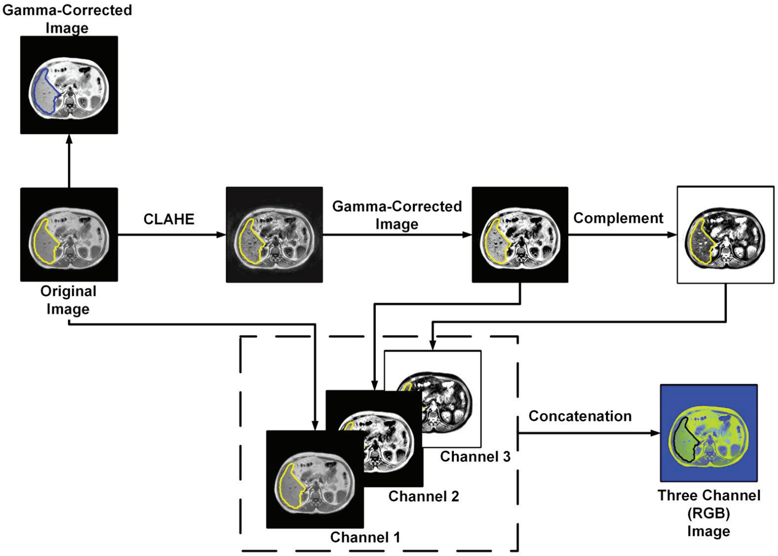


Figure 5. Visualization of image enhancement techniques.

2.3.1. Fold Creation

The methodology follows five-fold cross-validation techniques for validating the network performances. From the preprocessed dataset, five folds were created. In each fold training, validation, and testing set ratios were 70%, 10%, and 20%, which corresponds to 453, 65, and 129 DICOM slices, respectively. This was done to make sure that the performance metric represents the performance of the trained network on the complete dataset.

2.3.2. Augmentation

The training set for each fold was augmented using geometrical spatial transformation of coordinates (rotation and translation). Geometric spatial transformations represent a widely recognized and efficient technique for processing topographic imaging datasets [40,41].

The affine matrix for rotation $I_{rotation}$ and for translation $I_{translation}$ can be denoted as

$$I_{rotation} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$I_{translation} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \tag{6}$$

where the values of θ are defined by the set:

$$\theta = \{\pm 5^\circ, \pm 10^\circ, \pm 15^\circ, \pm 20^\circ, ..., \pm 90^\circ\} \tag{7}$$

and the values (t_x, t_y) are defined by the set:

$$(t_x, t_y) = \{(-10, 10), (+10, -10), (-10, +10), (10, 10)\} \tag{8}$$

The validation and testing sets were not augmented. After augmentation, each training fold consisted of around 6700 slices. The validation set was used to avoid overfitting, which is a common problem in machine learning model development [42,43].

2.3.3. Image Enhancement

Image enhancement includes gamma correction for each fold. For each of the pixels $f(x,y)$, the gamma correction can be denoted as [44]

$$g(x,y) = 255\left(\frac{f(x,y)}{255}\right)^{\frac{1}{\lambda}} \tag{9}$$

where g(x,y) denotes the gamma corrected pixel value, and the value of λ is considered to be 0.5 in this study. And, for all $f(x,y) > 200$, $f(x,y)$ is considered to be 255 to enhance the targeted region. Another image enhancement technique called contrast-limited adaptive histogram equalization (CLAHE) was used in the three-channel (or RGB) image construction technique. If in a histogram kth, the intensity value is r_k, and the number of pixels with the r_k intensity value is n_k, then for an $M \times N$ dimensional image, the equalized histogram can be represented by

$$p(r_k) = \frac{n_k}{M \times N} \tag{10}$$

CLAHE is an adaptive histogram equalization technique that undergoes transformation over local regions. Here, a matrix of 8×8 dimension was used for local histogram equalization. The output histogram from the CLAHE transformed image follows the

Rayleigh distribution. Gamma correction was applied to the CLAHE-enhanced image, and finally, the image was complemented. The image compliment $f^{-1}(x)$ can be expressed as

$$f^{-1}(x) = 255 - f(x) \tag{11}$$

The three-channel (or RGB) image was constructed by concatenating the original image, the gamma-corrected CLAHE enhanced image, and the complement of the gamma-corrected CLAHE enhanced image.

2.4. Deep Neural Networks

UNet-like architectures with pretrained deep dense, residual, and inception encoder backbones previously showed high performance in both classification and segmentation tasks for 2D chest X-rays [45]. UNet++ with deep dense blocks showed benchmark performance in segmenting lung content from volumetric CT scans [46]. These segmentation networks also performed well in solving complex problems such as detecting intracranial hemorrhages [47]. These studies inspired us to investigate these UNet-like architectures with pretrained encoder backbones for liver segmentation tasks from MR scans. UNet, UNet++, and feature pyramid network (FPN) segmentation networks were investigated with varying depths of dense, residual, and inception encoder backbones. The network architectures are shown in Figure 6. The encoder backbones were pretrained dense, residual, and inception blocks (marked in light orange). The decoder (light blue) uses transpose convolution blocks for upscaling the vector output from the bottleneck (marked in dark blue) output to construct the segmentation mask. The yellow blocks in UNet++ and FPN represent both concatenation and convolution blocks.

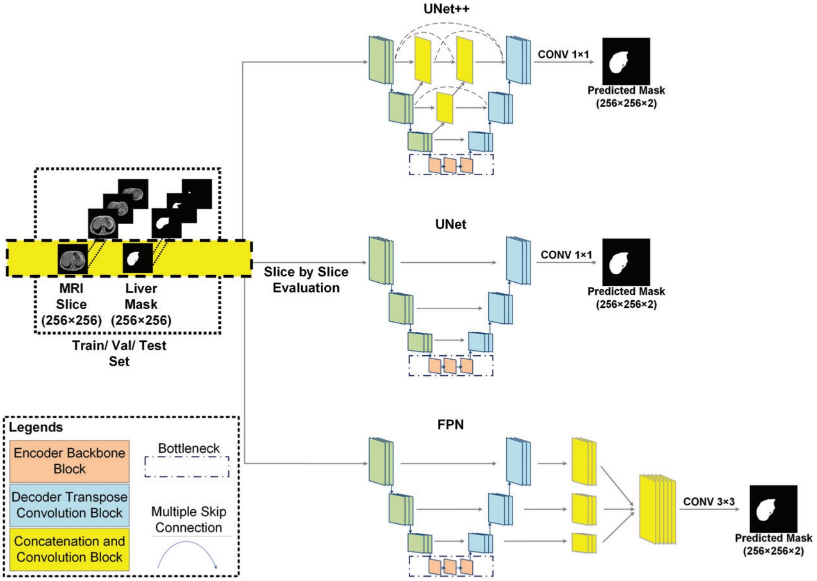

UNet++
CONV 1×1
Predicted Mask (256×256×2)
UNet
CONV 1×1
Predicted Mask (256×256×2)
FPN
CONV 3×3
Predicted Mask (256×256×2)
MRI Slice (256×256)
Liver Mask (256×256)
Train/ Val/ Test Set
Slice by Slice Evaluation
Legends
Encoder Backbone Block
Decoder Transpose Convolution Block
Concatenation and Convolution Block
Bottleneck
Multiple Skip Connection

Figure 6. Network architectures of different segmentation networks and investigation frameworks. The varying depth of pretrained dense, residual, and inception encoder backbones were investigated for UNet++, UNet, and FPN segmentation network architectures.

2.4.1. UNet

UNet architecture consists of an encoder and a decoder part. The encoder part reduces the input image size in each of the convolutional blocks through max pooling. In the final encoder block, the two-dimensional original image matrix is reduced to a vector array.

The decoder part upscales the converted vector array in each block through convolutional blocks and upconvolution layers. Lastly, the skip connections among encoder–decoder blocks transfer weights for localizing the region of interest. These skip connections are similar to the attention mechanism [48].

2.4.2. UNet++

UNet++ is an extension of the UNet and wide UNet architecture. It utilizes the concept of deep supervision. UNet++ also introduces nested convolutional blocks inside each skip pathway, and such blocks enhance the quality of feature spaces that are passed to the decoder blocks [49,50].

2.4.3. Feature Pyramid Network (FPN)

In the FPN network, weight connections from the UNet decoder blocks are fed through skip connections to feature pyramid blocks. Further, the output from each feature pyramid block is fed into a single convolutional block. Finally, the output from the convolutional block is fed into a rectilinear unit (ReLU) activation layer for generating the predicted masks [51].

2.4.4. Pretrained Backbones

The concept of transfer learning is utilized to enhance the segmentation performance and reduce the training time. Several pretrained encoders (variants of dense, residual, and inception networks), which were trained on the ImageNet computer vision database [52], were used as the backbones. For each backbone variant, three varying depths were investigated. The variants of DenseNets were DenseNet201, DenseNet161, and DenseNet121 [52–54], while the variants of residual networks were ResNet152, ResNet50, and ResNet18 [55]. InceptionV4 and InceptionResNet were the variants of the inception backbones [56].

2.5. Experiments

Two major experiments were carried out in this study: (i) the generalized model and (ii) the specialized network for handling anatomical ambiguity.

2.5.1. Generalized Model

In this experiment, MRI slices with different liver sizes were used in training and evaluation and the model was generally not specific to any particular liver size. Then, the effects of image enhancement on the generalized model were investigated. A total of 24 networks (three architectures with eight backbones) were tested on three versions of MRI images (i.e., original, gamma-corrected, and 3-channel view) to segment the liver.

2.5.2. Specialized Network for Handling Anatomical Ambiguity

To enhance the performance of the segmentation network in segmenting the liver region from the MRI slices where the liver shape varies, multiple segmentation networks needed to be trained to segment the liver region reliably. A total of 90 slices from the inferior part of the liver and the upper right pole of the kidney were trained separately. Exact preprocessing and processing steps were followed for this set of slices, which was discussed previously. For this specific task, only varying depths of the ResNet encoder–decoder backbones with UNet++, UNet, and FPN were investigated as the ResNet showed better performance in the preliminary study. Three variants of ResNet and Inception-ResnetV2 with three architectures (a total of 12 experiments) were investigated specifically for the slices with small liver contents.

2.5.3. Cascaded Network

Since the liver size varies in the MRI volume, every single generalized model proposed in the literature fails to generalize. Therefore, we propose a cascaded model using a decision function to improve the performance of the segmentation network. The architecture of the cascaded network is depicted in Figure 7. The volumetric MRI scan is fed into the network slice by slice. At first, a liver mask is predicted from the generalized network. From the first predicted mask, the number of predicted white pixels is calculated. The following equation is used to decide the potential shape of the liver mask in the slice under investigation, where k represents the white pixel count:

$$Liver\_Content = \begin{cases} Absent, \text{ if } k = 0 \\ Small, \text{ if } 1 \leq k \leq 750 \\ large, \text{ if } k > 750 \end{cases} \tag{12}$$

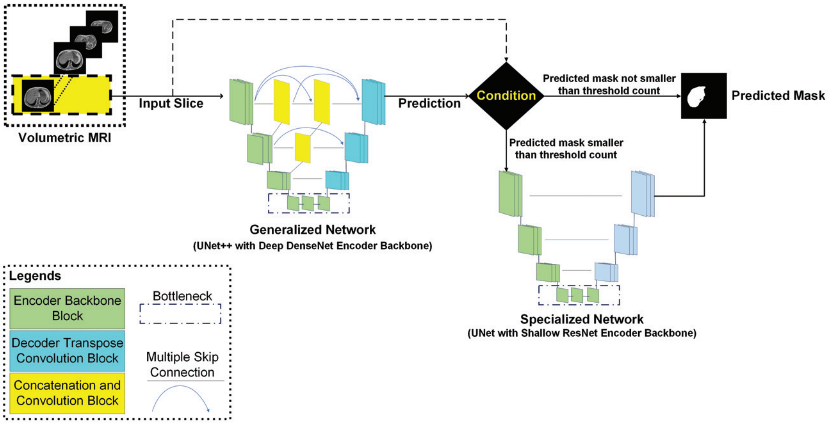

Volumetric MRI
Input Slice
Prediction
Condition
Predicted mask not smaller than threshold count
Predicted Mask
Predicted mask smaller than threshold count
Generalized Network
(UNet++ with Deep DenseNet Encoder Backbone)
Specialized Network
(UNet with Shallow ResNet Encoder Backbone)
Legends
Encoder Backbone Block
Decoder Transpose Convolution Block
Concatenation and Convolution Block
Bottleneck
Multiple Skip Connection

Figure 7. Cascaded network for handling anatomical ambiguity: the generalized network predicts an initial mask; if the pixel count for the predicted mask refers to a constrained or null liver content, then the input slice is fed into the specialized network.

If the number of white pixels is zero, there is no liver in the slice and so the mask is completely black. However, if the decision function identifies a number between 1–750, the slice is again fed into the specialized network for producing the final mask. However, if the number is higher than 750, the mask generated by the generalized model is used as the final liver mask.

The sets of large, small, and absent liver contents are created on the basis of the topographic visualization of the abdominal anatomy, which was described earlier in Section 2.2.1. The liver content is maximum in the axial views from the middle part of the liver. Moreover, the liver content is medium and constrained in the axial views from the superior part of the liver and the inferior part of the liver, respectively. In the axial view from the upper part of the kidney, the liver content is absent. In this perspective, the set of large liver content is constructed with the axial views from the middle part and superior part of the liver. The axial views from the inferior part of the liver are represented in the set of small liver content. Lastly, the set of absent liver content is formed by the axial views from the upper part of the right kidney. The threshold values are then determined by analyzing the pixel counts in each of the sets.

Generally, the axial views from the superior part and inferior part of the liver have significant liver content and the liver area can be comfortably segmented. However, ambiguity arises for the axial views from the inferior part of the liver and the upper part of the right kidney, as the liver portion is significantly constrained. In such a perspective, segmentation performance may improve if slices from these two complicated axial views are handled with a separate network, which is only trained with such cases. Thus, such a cascaded approach was investigated.

2.6. Loss Function

Binary cross-entropy (BCE) loss is typically used for classification tasks. As any semantic segmentation task can be considered as a classification task at the pixel level, this loss is also effective for segmentation. BCE loss can be expressed by [57,58]

$$Loss_{(BCE)} = \frac{1}{N} \sum_{i=0}^{N-1} -(y_i \log(\hat{y}_i) + (1-y)\log(1-\hat{y}_i)) \tag{13}$$

The dice coefficient is used to calculate the similarity index between ground truth and predicted masks for segmentation tasks. Dice loss is a region-based loss function and it is introduced in [59]. Dice loss can be expressed by

$$Loss_{(DICE)} = 1 - \frac{\sum\limits_{i=0}^{N-1} y_i \hat{y}_i}{\sum\limits_{i=0}^{N-1} y_i^2 + \sum\limits_{i=0}^{N-1} \hat{y}_i^2 + \epsilon} \tag{14}$$

In Equations (13) and (14), N represents the total number of pixels, y_i represents the ith pixel in the ground truth mask, and $\hat{y}_i$ represents the ith pixel

Initially, both the mentioned loss functions were investigated to find the optimum solution. However, the detailed investigation was carried out with the BCE loss, as it demonstrated superior performance over dice loss in the initial investigation.

2.7. Training Parameters

In order to conduct a uniform comparison among the network performances, it was indispensable to use the same training parameters for all the networks. All the training was conducted in an NVIDIA Tesla P100-PCIE graphics processing unit (GPU) with 16 gigabytes (GB) og memory. The initial learning rate was set to 0.0001 with a learning factor LR of 0.02. If the validation loss did not show significant changes in 10 epochs, the learning rate was reduced by $\frac{1}{LR}$. The maximum epoch number was set to be 100 for each fold, but if the validation loss was constant for 20 epochs, the training was terminated. Z-score normalization was used, which uses the standard deviation and mean of the raw MRI slices for normalizing each image. For optimizing the process of gradient descent, in each of the epochs, the ADAM optimization algorithm was used as it showed superior performance over stochastic gradient descent (SGD) in the initial investigation [60,61].

2.8. Evaluation Metrics

For evaluating the performance of each investigated network, the accuracy, dice similarity coefficient (DSC) (i.e., F1-score), and intersection of union (IoU) were computed. For each network, the average metrics for each ground truth mask and predicted mask pairs were calculated. Accuracy, DSC, and IoU can be expressed by

$$Accuracy = \frac{TP + TN}{2 \times TP + TN + FP + FN} \tag{15}$$

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{16}$$

$$IoU = \frac{TP}{2 \times TP + FP + FN} \tag{17}$$

In Equations (15), (16), and (17), TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

2.9. Cloud Deployment

A cloud-based application for real-time liver segmentation from MRI images was deployed. The deep learning model was deployed in the cloud back-end server, which runs on an 8-core, 32 GB Memory Apache Linux instance hired from Google Cloud Perform (GCP). The back-end server was connected to a SQL database for storing the MRI images. The application is cross-platform compatible and users can access the application anytime via a web browser from any edge device. The cloud-based application can be remotely connected with a Picture Archiving System (PACS) for assisting radiologists in liver pathology investigations. In order to provide more convenient remote access for clinicians, an Android application was also developed. Figure 1 superficially describes the cloud application. To ensure the robustness of the segmentation network, an automated self-learning scheduler was implemented in the back-end server following the concept discussed [62]. The scheduler automatically retrains the deployed model with the incoming new data provided by the user, and such an approach boosts the network's performance on unseen real-world data.

3. Results and Discussion

The results from different investigations are discussed in this section. Later, a performance comparison is presented, which compares the efficiency of the proposed approach with the reported high-performance techniques in the literature for liver segmentation on the same MRI test set. In the following section, the performance of the generalized model, specialized model, and cascaded models are presented.

3.1. Generalized Model

The Table 2 summarizes the network performance for segmenting the liver region in the MR slices using a generalized model. Deep networks showed superiority in performance in comparison to shallow networks. On the nonenhanced images (original image), UNet++ with dense backbones showed the top performance. UNet++ with a DenseNet201 backbone showed the best performance with a DSC and IoU of 94.3% and 91.00%, respectively. On the nonenhanced images, UNet with different DenseNet backbones exhibited a similar performance.

For the three-channel image set, networks with DenseNet backbones demonstrated slightly better performance compared to other networks. Among the variants of the DenseNet model, DenseNet161 performed the best for this specific image set. Both FPN and UNet with DenseNet161 backbones achieved a DSC of over 93%. Among the investigated image enhancement techniques, the gamma-enhanced image set performed the worst.

The liver content is maximum in the slices showing the middle part of the liver and also significantly larger in the superior part of the liver. For these two specific types of slices, all of the DenseNet backbones showed excellent performances. Figure 8a shows the predicted liver masks for the slices from the middle part of the liver. The figure shows that all of the networks can segment the liver region accurately.

3.2. Effects of Image Enhancement for Generalized Model

It can be observed that the network performances were slightly decreased when image enhancement techniques were implemented (Table 2). This is due to the ambiguity that arises from the slices where the liver content varies widely. Though image enhancement was very effective for the slices where the liver portion was significant, the performance dropped when the liver size was minimum in the slice of investigation. Figure 8b shows such a sample liver slice with the ground truth mask and the masks predicted by different models.

Due to this ambiguity, finding a generalized image enhancement technique for such a complex and varying anatomy is very challenging.

3.3. Limitation of the Generalized Model

In the case of the slices of the middle part of the liver, all ResNet and inception backbones demonstrated satisfactory performance. Figure 9 shows the predicted masks from top-performing networks for the middle part of the liver (large liver content), inferior part of the liver (small liver content), and upper pole of the kidney (no liver content) for the original T_1-weighted images. Figure 9 shows that the generalized model performed well for the slices with large liver content and for the slices where the liver was absent. However, when the liver content was small, the generalized model struggled to locate the liver area precisely.

Table 2. Summary of the investigated network performances from the generalized approach. UNet++ with DenseNet201 encoder exhibited the best performance.

Networks		Original			Three Channel			Gamma Corrected		
Architecture	**Backbone**	**Acc. (%)**	**IoU (%)**	**DSC (%)**	**Acc. (%)**	**IoU (%)**	**DSC (%)**	**Acc. (%)**	**IoU (%)**	**DSC (%)**
UNet++	DenseNet201	99.73	91.00	94.30	99.60	88.95	92.35	99.42	89.28	91.91
	DenseNet161	99.68	89.78	93.06	99.71	89.60	92.95	99.66	87.00	90.30
	DenseNet121	99.43	89.17	92.57	99.66	90.08	93.40	99.56	87.58	90.92
	ResNet152	99.70	89.79	93.13	99.67	87.97	91.34	99.70	89.00	92.33
	ResNet50	99.70	90.42	93.81	99.68	89.54	92.98	99.66	88.13	91.64
	ResNet18	99.70	89.73	93.08	99.70	89.63	93.01	99.63	84.55	88.50
	Inception-resnet-v2	99.71	89.16	92.57	99.65	88.31	92.10	99.70	89.60	91.98
	inception-v4	99.70	87.98	91.29	99.68	89.23	92.62	99.70	89.79	92.16
UNet	DenseNet201	99.76	89.98	93.22	99.72	88.77	92.13	99.78	88.74	92.18
	DenseNet161	99.57	90.48	93.84	99.43	90.08	93.60	99.45	87.58	90.92
	DenseNet121	99.43	89.88	93.27	99.66	89.48	92.90	99.64	87.04	90.31
	ResNet152	99.69	89.46	92.97	99.67	88.66	92.25	99.67	88.91	92.35
	ResNet50	99.68	87.48	90.93	99.66	85.49	89.01	99.68	88.79	92.36
	ResNet18	99.67	88.16	91.83	99.67	86.83	90.38	99.68	88.77	92.31
	Inception-resnet-v2	99.66	87.68	91.41	99.68	88.20	91.80	99.70	87.81	91.32
	inception-v4	99.68	88.64	92.34	99.70	90.68	93.47	99.62	87.89	91.71
FPN	DenseNet201	99.65	89.45	92.83	99.36	89.50	92.97	99.47	88.33	91.87
	DenseNet161	99.70	89.38	92.77	99.66	89.32	93.00	99.53	88.11	91.85
	DenseNet121	99.47	87.52	91.08	99.47	89.39	92.94	99.71	86.91	90.49
	ResNet152	99.66	88.49	92.08	99.67	88.90	92.59	99.68	87.85	91.46
	ResNet50	99.69	89.01	92.52	99.65	88.15	91.88	99.66	88.76	92.40
	ResNet18	99.68	88.33	91.91	99.66	88.10	91.95	99.67	88.58	92.33
	Inception-resnet-v2	99.61	87.03	91.46	99.67	88.17	92.08	99.65	88.52	92.39
	inception-v4	99.62	85.52	90.85	99.66	88.64	92.55	99.66	88.64	92.55

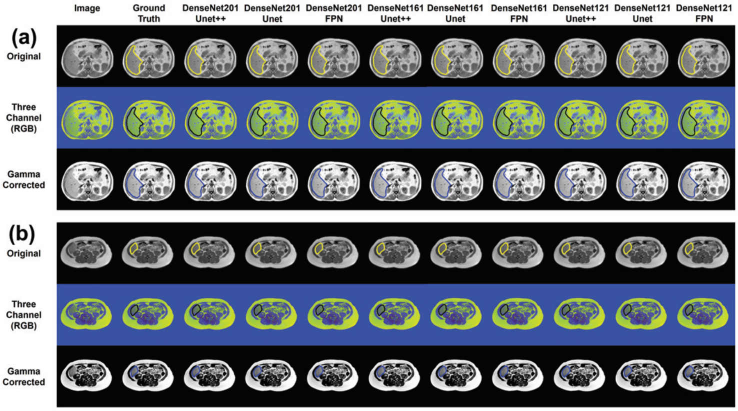


Figure 8. Visualization of the predicted masks from the networks with DenseNet backbones for a sample axial slice showing the middle part of the liver (**a**) and the inferior part of the liver (**b**).

A more detailed picture is shown in Table 3. The table illustrates the fold-wise slice distribution and the observed DSC for each of the different groups of liver shapes for the top-performing UNet++ model with the DenseNet201 backbone. Though the percentage of slices from the middle part of the liver was the minimum for all the folds, the DSC value for the best-performing model was still over 95% for each fold. Slices with medium liver content occurred the most and the DSC value for each fold was around 95%. The network also efficiently handled slices where the liver content is absent.

However, the model performance was greatly reduced for the slices where liver content was small. It is worth mentioning that the number of such slices in the training set was also insignificant. Our hypothesis was that handling such slices by a separate model may improve the overall segmentation performance, which is explored in this study.

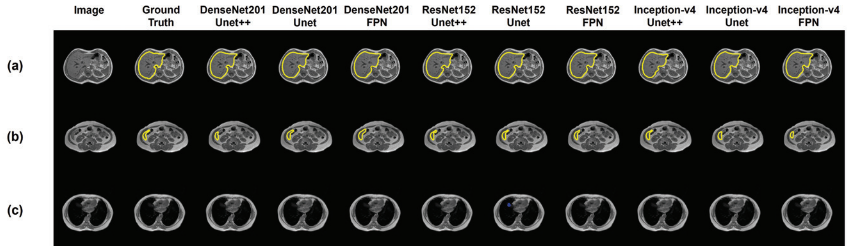


Figure 9. Visualization of the predicted masks from selected networks for a sample slice in (**a**) middle part of the liver (large liver content), (**b**) inferior part of the liver (small liver content), and (**c**) upper pole of the kidney (no liver content).

Table 3. Distribution of slices of distinct axial views in train set and test set, along with the observed DSC. Slices depicting axial view from the inferior part of the liver holds a constrained liver content and exhibits anatomical ambiguity.

Fold No	Middle Part of Liver (Liver Content: Large)			Superior Part of Liver (Liver Content: Medium)			Inferior Part of Liver (Liver Content: Small)			Upper Part of Kidney (Liver Content: Absent)		
	Train Set Slice %	Test Set Slice %	DSC (%)	Train Set Slice %	Test Set Slice %	DSC (%)	Train Set Slice %	Test Set Slice %	DSC (%)	Train Set Slice %	Test Set Slice %	DSC (%)
1	7.74%	19.38%	97.03%	45.63%	34.89%	95.33%	12.71%	16.28%	81.95%	34.12%	29.46%	100.00%
2	7.73%	11.63%	95.75%	44.36%	41.86%	95.11%	13.25%	11.63%	82.64%	34.66%	34.88%	95.55%
3	6.69%	17.83%	96.17%	43.43%	41.86%	95.20%	11.79%	12.40%	78.13%	35.85 %	30.23%	97.43%
4	7.67%	10.85%	95.70%	43.26%	42.63%	95.23%	14.90%	9.30%	80.20%	34.18%	37.21%	97.91%
5	6.86%	16.79%	97.35%	44.25%	38.17%	93.90%	14.15%	9.16%	82.46%	34.75%	35.88%	94.78%

3.4. *Specialized Network for Handling Anatomical Ambiguity*

Table 4 illustrates the performance of the specialized models trained with different architectures and different backbones in comparison to the best-performing generalized model in segmenting the MR slices with a small liver content.

The UNet with ResNet18 encoder backbone showed superior performance over the other investigated networks, with an IoU and DSC of 77.00% and 86.22%, respectively. For UNet++, the Inception-resnet-V2 encoder backbone showed better performance over the varying depths of ResNet backbones with an IoU and DSC of 75.58% and 84.03%, respectively. The shallow ResNet18 backbone performed better for the FPN architecture over other pretrained encoder backbones with an IoU and DSC of 71.20% and 82.04%, respectively. Each of the top-performing encoder backbones for UNet, UNet++, and FPN performed better than the top-performing generalized network for this task, which demonstrated an IoU and DSC of 70.74% and 80.88%, respectively. Lastly, the shallow networks performed better compared to deep networks for this specific task, as the liver content in the slice was small.

3.5. Cascaded Network

Figure 10 shows the predicted masks from the proposed cascaded network. It can be observed that such an approach enhances the mask quality for slices with a small liver content. Combining both the generalized and specialized network enhances the performance of the network for segmenting the liver region. Table 5 summarizes the performance metrics for the generalized network and the cascaded network. Cascading both the networks improves the overall DSC score (from 94.3 to 95.15%).

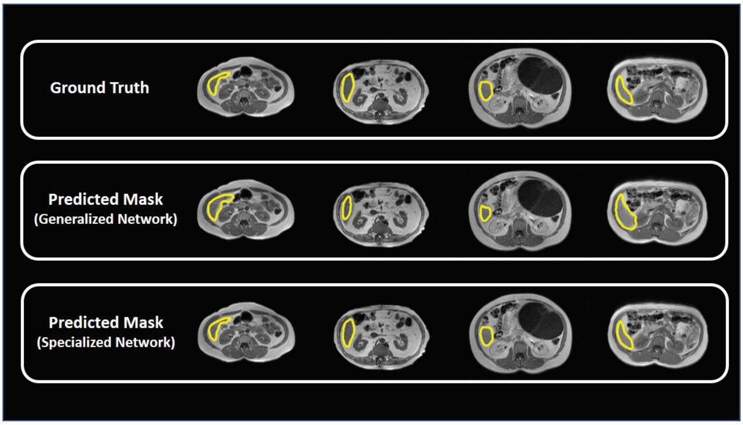


Figure 10. Comparison of the predicted masks from the generalized and specialized networks for sample MR slices with small liver content.

Table 4. Summary of the investigated network performance for the slices with small liver content using different specialized models and the best-performing generalized model. UNet with ResNet18 backbone showed improved performance for the task.

Networks		Metrics (Specialized Network)			Metrics (Best-Performing Generalized Network)		
Architecture	**Backbone**	**Acc.** (%)	**IoU** (%)	**DSC** (%)	**Acc.** (%)	**IoU** (%)	**DSC** (%)
UNet	ResNet18	99.64	77.00	86.22	99.76	70.74	80.88
	ResNet50	99.81	72.06	80.94			
	ResNet152	99.78	70.00	79.73			
	Inception-resnet-v2	99.70	72.73	81.72			
UNet++	ResNet18	99.78	71.71	78.38			
	ResNet50	99.76	71.62	80.96			
	ResNet152	99.80	71.89	81.02			
	Inception-resnet-v2	99.78	75.58	84.03			
FPN	ResNet18	99.80	71.20	82.04			
	ResNet50	99.77	69.75	79.77			
	ResNet152	99.78	71.86	81.20			
	Inception-resnet-v2	99.80	70.92	80.41			

Table 5. Performance metrics for the best-performing generalized and cascaded network. Here, the results of the cascaded network are marked in gray.

Experiments	**Acc.** (%)	**IoU** (%)	**DSC** (%)
Generalized Network	99.73%	91.00%	94.30%
Cascaded Network	99.70%	92.10%	95.15%

3.6. Discussion

The performance comparison of our proposed framework with all of the existing high-performance networks (using the same test for evaluation) is summarized in Table 6. X. Zhong et al. [25] investigated deep action learning for abdominal organ segmentation tasks from volumetric MRI images. Their proposed network demonstrated superiority over 3D UNet in terms of overall performance, and achieved a DSC of 80.6% for the liver segmentation task. P. Pandey et al. [26] explored a contrastive semisupervised approach for the same task, and it achieved a DSC of 85.9%. The proposed method generates patches for each slice, which enhances the feature space. Mitta et al. [28] achieved a DSC of 88.12% on the test set by using W-Net with attention gates. J. Hong et al. [29] and X. Wang et al. [30] used source-free unsupervised learning and bidirectional searching for the segmentation task, respectively. By using geometric edge enhancement, S. Mulay et al. [31] boosted the performance of the mask R-CNN for the liver segmentation task on this test set. L. Zbinden et al. [32] achieved a DSC of 93.60% by implementing nnUNet on T_1-weighted MRI slices.

Our proposed cascaded framework outperforms all of these existing high-performance techniques by a large margin with a DSC of 95.15%. As discussed previously, the size of the liver content in an arbitrary MRI slice depends on its axial view source. Any generalized segmentation network can perform comparatively better when the liver content is significant in the given MRI slice (axial view from the middle part of the liver). On the contrary, the network faces ambiguity when the liver content is reduced for the given MRI slice (axial view from the upper pole of the kidney, the inferior pole of the kidney, and the superior part of the liver). As a result, the network performance drops significantly for these specific groups of slices where the liver content is small. This specific cause for reduced segmentation performance is overlooked in all of the previous studies. Our proposed framework separately handles this specific group of slices with a small liver content, which generates ambiguity through a specialized network, thus enhancing the overall segmentation performance.

Table 6. Comparison of the proposed method (marked in gray) with existing studies that used the same testing set.

Authors	Methodology and Approach	Metric (DSC)
X. Zhong et al. [25]	Deep action learning with 3D UNet	80.60 ± 5.30%
P. Pandey et al. [26]	Contrastive Semi Supervised Learning Approach with UNet	85.90%
D. Mitta et al. [28]	W-Net with attention gates	88.12%
J. Hong et al. [29]	Source Free Unsupervised UNet	88.40%
X. Wang et al. [30]	Bidirectional Searching Neural Net	89.80%
S. Mulay et al. [31]	Mask R-CNN	80.00%
	Geomatric Edge Enhancement based Mask R-CNN	91.00%
L. Zbinden et al. [32]	nnUNet	93.60%
Proposed	Cascaded Network for Handling Anatomical Ambiguity	95.15%

4. Conclusions

Abdominal organ segmentation is a challenging task due to the complexity of the anatomy of the abdominal area and the close proximity of multiple organs. Ambiguity in the segmentation of the liver arises due to the variance in its anatomical shape in the MRI volume. The MRI modality is favored by clinicians for liver pathology diagnosis. However, automated liver segmentation from MRI scans is a demanding task. In this research, we proposed a novel cascaded network for liver segmentation from T_1-weighted MR images. The proposed network treats each axial view distinctly and achieved a DSC of 95.15% on the publicly available CHAOS MRI dataset. Such an approach can also be investigated for other abdominal organ segmentation tasks, such as those involving the kidneys and spleen. The proposed network was also deployed as an open-source application in a cloud server for demonstration purposes. This application can later be integrated with PACS for clinical usage. Lastly, we also investigated the effects of different image enhancement techniques for liver segmentation tasks from MR scans.

Author Contributions: Conceptualization, M.S.A.H., M.E.H.C., M.S.K. and M.H.; methodology, M.S.A.H., M.E.H.C., S.G., M.S.K., E.H.B., M.H. and A.S.; software, M.S.I.S., M.S.A.H., S.G., M.A.A., A.A. and S.M.; validation, M.S.A.H., A.K., I.A.-H. and E.H.B.; formal analysis, M.S.A.H.; investigation, M.S.A.H., E.H.B., M.H., I.A.-H. and M.E.H.C.; resources, M.S.K. and A.K.; data curation, M.S.A.H., S.G., A.S., S.M. and I.A.-H.; writing—original draft preparation, M.S.A.H., S.G. and S.M.; writing—review and editing, all authors; visualization, M.S.A.H., M.S.I.S., A.K., M.A.A. and A.A.;

supervision, M.E.H.C., M.S.K., A.A., M.A.A. and M.H.; project administration, M.E.H.C., M.S.K., A.A. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Qatar University High Impact grant QUHI-CENG-23/24-216 and student grant QUST-1-CENG-2023-796 and is also supported via funding from Prince Sattam Bin Abdulaziz University project number (PSAU/2023/R/1444). The statements made herein are solely the responsibility of the authors. The open-access publication cost is covered by the Qatar National Library.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This article is based on previously conducted studies and does not contain any studies with human participants or animals. The dataset used in this research is open access and can be accessed from https://chaos.grand-challenge.org/Data/, accessed on 9 August 2023. Kavuar et al. [27] followed standard protocol for creating this open-access dataset.

Acknowledgments: The authors would like to thank Kavuar et al. [27] for sharing the dataset publicly which make the research possible.

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

References

1. Aggarwal, R.; Sounderajah, V.; Martin, G.; Ting, D.S.; Karthikesalingam, A.; King, D.; Ashrafian, H.; Darzi, A. Diagnostic accuracy of deep learning in medical imaging: A systematic review and meta-analysis. *NPJ Digit. Med.* **2021**, *4*, 65. [CrossRef] [PubMed]
2. Barragán-Montero, A.; Javaid, U.; Valdés, G.; Nguyen, D.; Desbordes, P.; Macq, B.; Willems, S.; Vandewinckele, L.; Holmström, M.; Löfman, F.; et al. Artificial intelligence and machine learning for medical imaging: A technology review. *Phys. Med.* **2021**, *83*, 242–256. [CrossRef]
3. Chen, Z.; Song, Y.; Chang, T.-H.; Wan, X. Generating Radiology Reports via Memory-Driven Transformer. *arXiv* **2020**, arXiv:2010.16056.
4. Tahir, A.M.; Chowdhury, M.E.; Khandakar, A.; Rahman, T.; Qiblawey, Y.; Khurshid, U.; Kiranyaz, S.; Ibtehaz, N.; Rahman, M.S.; Al-Maadeed, S.; et al. COVID-19 infection localization and severity grading from chest X-ray images. *Comput. Biol. Med.* **2021**, *139*, 105002. [CrossRef] [PubMed]
5. Abbas, T.O.; AbdelMoniem, M.; Khalil, I.; Hossain, M.S.A.; Chowdhury, M.E. Deep learning based automated quantification of urethral plate characteristics using the plate objective scoring tool (POST). *arXiv* **2023**, arXiv:2209.13848.
6. Dou, Q.; Chen, H.; Jin, Y.; Yu, L.; Qin, J.; Heng, P.A. 3D Deeply Supervised Network for Automatic Liver Segmentation from CT Volumes. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, 17–21 October 2016; Proceedings, Part II 19; Springer: Cham, Switzerland, 2016. [CrossRef]
7. Chen, C.; Zhou, K.; Zha, M.; Qu, X.; Guo, X.; Chen, H.; Wang, Z.; Xiao, R. An effective deep neural network for lung lesions segmentation from COVID-19 CT images. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6528–6538. [CrossRef]
8. Chen, C.; Xiao, R.; Zhang, T.; Lu, Y.; Guo, X.; Wang, J.; Chen, H.; Wang, Z. Pathological lung segmentation in chest CT images based on improved random walker. *Comput. Methods Programs Biomed.* **2021**, *200*, 105864. [CrossRef]
9. Hu, P.; Wu, F.; Peng, J.; Liang, P.; Kong, D. Automatic 3D liver segmentation based on deep learning and globally optimized surface evolution. *Phys. Med. Biol.* **2016**, *61*, 8676. [CrossRef]
10. Liu, X.; Song, L.; Liu, S.; Zhang, Y. A review of deep-learning-based medical image segmentation methods. *Sustainability* **2021**, *13*, 1224. [CrossRef]
11. Yang, R.; Yu, Y. Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis. *Front. Oncol.* **2021**, *11*, 638182. [CrossRef]
12. Liu, L.; Wolterink, J.M.; Brune, C.; Veldhuis, R.N.J. Anatomy-aided deep learning for medical image segmentation: A review. *Phys. Med. Biol.* **2021**, *66*, 11TR01. [CrossRef] [PubMed]
13. Peng, J.; Wang, Y.; Kong, D. Liver segmentation with constrained convex variational model. *Pattern Recognit. Lett.* **2014**, *43*, 81–88. [CrossRef]
14. Liu, Z.; Song, Y.Q.; Sheng, V.S.; Wang, L.; Jiang, R.; Zhang, X.; Yuan, D. Liver CT sequence segmentation based with improved U-Net and graph cut. *Expert Syst. Appl.* **2019**, *126*, 54–63. [CrossRef]
15. Tang, X.; Jafargholi Rangraz, E.; Coudyzer, W.; Bertels, J.; Robben, D.; Schramm, G.; Deckers, W.; Maleux, G.; Baete, K.; Verslype, C.; et al. Whole liver segmentation based on deep learning and manual adjustment for clinical use in SIRT. *Eur. J. Nucl. Med. Mol. Imag.* **2020**, *47*, 2742–2752. [CrossRef]
16. Kim, K.; Chun, J. A new hyper parameter of hounsfield unit range in liver segmentation. *J. Internet Comput. Serv.* **2020**, *21*, 103–111. [CrossRef]
17. Xiang, K.; Jiang, B.; Shang, D. The overview of the deep learning integrated into the medical imaging of liver: A review. *Hepatol. Int.* **2021**, *15*, 868–880. [CrossRef]

18. Elbanna, K.Y.; Kielar, A.Z. Computed Tomography Versus Magnetic Resonance Imaging for Hepatic Lesion Characterization/Diagnosis. *Clin. Liver Dis.* **2021**, *17*, 159. [CrossRef]
19. Coenegrachts, K. Magnetic resonance imaging of the liver: New imaging strategies for evaluating focal liver lesions. *World J. Radiol.* **2009**, *1*, 72. [CrossRef]
20. Caseiro-Alves, F.; Brito, J.; Araujo, A.E.; Belo-Soares, P.; Rodrigues, H.; Cipriano, A.; Sousa, D.; Mathieu, D. Liver haemangioma: Common and uncommon findings and how to improve the differential diagnosis. *Eur. Radiol.* **2007**, *17*, 1544–1554. [CrossRef]
21. Wang, G.; Zhu, S.; Li, X. Comparison of values of CT and MRI imaging in the diagnosis of hepatocellular carcinoma and analysis of prognostic factors. *Oncol. Lett.* **2019**, *17*, 1184–1188. [CrossRef]
22. Gibbs, J.F.; Litwin, A.M.; Kahlenberg, M.S. Contemporary management of benign liver tumors. *Surg. Clin. N. Am.* **2004**, *84*, 463–480. [CrossRef]
23. Mostafa, A.; Hassanien, A.E.; Houseni, M.; Hefny, H. Liver segmentation in MRI images based on whale optimization algorithm. *Multimed. Tools Appl.* **2017**, *76*, 24931–24954. [CrossRef]
24. Hänsch, A.; Chlebus, G.; Meine, H.; Thielke, F.; Kock, F.; Paulus, T.; Abolmaali, N.; Schenk, A. Improving automatic liver tumor segmentation in late-phase MRI using multi-model training and 3D convolutional neural networks. *Sci. Rep.* **2022**, *12*, 12262. [CrossRef]
25. Zhong, X.; Amrehn, M.; Ravikumar, N.; Chen, S.; Strobel, N.; Birkhold, A.; Kowarschik, M.; Fahrig, R.; Maier, A. Deep action learning enables robust 3D segmentation of body organs in various CT and MRI images. *Sci. Rep.* **2021**, *11*, 3311. [CrossRef]
26. Pandey, P.; Pai, A.; Bhatt, N.; Das, P.; Makharia, G.; Ap, P. Contrastive semi-supervised learning for 2D medical image segmentation. *arXiv* **2021**. arXiv:2106.06801.
27. Kavur, A.E.; Gezer, N.S.; Barış, M.; Aslan, S.; Conze, P.H.; Groza, V.; Pham, D.D.; Chatterjee, S.; Ernst, P.; Özkan, S.; et al. CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation. *Med. Image Anal.* **2021**, *69*, 101950. [CrossRef]
28. Mitta, D.; Chatterjee, S.; Speck, O.; Nürnberger, A. Upgraded w-net with attention gates and its application in unsupervised 3d liver segmentation. *arXiv* **2020**, arXiv:2011.10654.
29. Hong, J.; Zhang, Y.D.; Chen, W. Source-free unsupervised domain adaptation for cross-modality abdominal multi-organ segmentation. *Knowl. Based Syst.* **2022**, *250*, 109155. [CrossRef]
30. Wang, X.; Xiang, T.; Zhang, C.; Song, Y.; Liu, D.; Huang, H.; Cai, W. Bix-Nas: Searching Efficient Bi-Directional Architecture for Medical Image Segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, 27 September–1 October 2021; Proceedings, Part I 24; Springer International Publishing: Cham, Switzerland, 2021; pp. 229–238. [CrossRef]
31. Mulay, S.; Deepika, G.; Jeevakala, S.; Ram, K.; Sivaprakasam, M. Liver Segmentation from Multimodal Images Using HED-Mask R-CNN. In Proceedings of the Multiscale Multimodal Medical Imaging: First International Workshop, MMMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, 13 October 2019; Proceedings 1; Springer International Publishing: Cham, Switzerland, 2019; pp. 68–75. [CrossRef]
32. Zbinden, L.; Catucci, D.; Suter, Y.; Berzigotti, A.; Ebner, L.; Christe, A.; Obmann, V.C.; Sznitman, R.; Huber, A.T. Convolutional neural network for automated segmentation of the liver and its vessels on non-contrast T1 vibe Dixon acquisitions. *Sci. Rep.* **2022**, *12*, 22059. [CrossRef]
33. Netter, F.H. *Section 4: Atlas of Human Anatomy*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2003; pp. 239–338. [CrossRef]
34. Suetens, P. Chapter 4—Magnetic Resonance Imaging. In *Fundamentals of Medical Imaging*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2017; pp. 64–104. [CrossRef]
35. de Bazelaire, C.M.; Duhamel, G.D.; Rofsky, N.M.; Alsop, D.C. MR imaging relaxation times of abdominal and pelvic tissues measured in vivo at 3.0 T: Preliminary results. *Radiology* **2004**, *230*, 652–659. [CrossRef]
36. Dimakis, N. Chapter 5—Magnetic Resonance Imaging (MRI). In *Introduction to Medical Imaging—Physics, Engineering and Clinical Applications*, 1st ed.; Cambridge University Press: Cambridge, UK, 2011; pp. 204–273. [CrossRef]
37. Rahman, T.; Chowdhury, M.E.; Khandakar, A.; Mahbub, Z.B.; Hossain, M.S.A.; Alhatou, A.; Abdalla, E.; Muthiyal, S.; Islam, K.F.; Kashem, S.B.A.; et al. BIO-CXRNET: A robust multimodal stacking machine learning technique for mortality risk prediction of COVID-19 patients using chest X-ray images and clinical data. *Neural Comput. Appl.* **2023**, *35*, 17461–17483. [CrossRef]
38. Hossain, S.A.; Rahman, M.A.; Chakrabarty, A.; Rashid, M.A.; Kuwana, A.; Kobayashi, H. Emotional State Classification from MUSIC-Based Features of Multichannel EEG Signals. *Bioengineering* **2023**, *10*, 99. [CrossRef]
39. Hossain, S.A.; Rahman, M.A.; Chakrabarty, A. MUSIC Model Based Neural Information Processing for Emotion Recognition from Multichannel EEG Signal. In Proceedings of the 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 26–27 August 2021; pp. 955–960. [CrossRef]
40. Nalepa, J.; Marcinkiewicz, M.; Kawulok, M. Data augmentation for brain-tumor segmentation: A review. *Front. Comput. Neurosci.* **2019**, *13*, 83. [CrossRef]
41. Safdar, M.F.; Alkobaisi, S.S.; Zahra, F.T. A Comparative Analysis of Data Augmentation Approaches for Magnetic Resonance Imaging (MRI) Scan Images of Brain Tumor. *Acta Inform. Med.* **2020**, *28*, 29–36. [CrossRef]
42. Islam, K.R.; Kumar, J.; Tan, T.L.; Reaz, M.B.I.; Rahman, T.; Khandakar, A.; Abbas, T.; Hossain, M.S.A.; Zughaier, S.M.; Chowdhury, M.E.H. Prognostic Model of ICU Admission Risk in Patients with COVID-19 Infection Using Machine Learning. *Diagnostics* **2022**, *12*, 2144. [CrossRef]

43. Mahmud, S.; Ibtehaz, N.; Khandakar, A.; Rahman, M.S.; Gonzales, A.J.; Rahman, T.; Hossain, M.S.; Hossain, M.S.A.; Faisal, M.A.A.; Abir, F.F.; et al. NABNet: A Nested Attention-guided BiConvLSTM network for a robust prediction of Blood Pressure components from reconstructed Arterial Blood Pressure waveforms using PPG and ECG signals. *Biomed. Signal Process. Control* **2023**, *79*, 104247. [CrossRef]
44. TRahman, T.; Khandakar, A.; Qiblawey, Y.; Tahir, A.; Kiranyaz, S.; Kashem, S.B.A.; Islam, M.T.; Al Maadeed, S.; Zughaier, S.M.; Khan, M.S.; et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Comput. Biol. Med.* **2021**, *132*, 104319. [CrossRef]
45. Rahman, T.; Khandakar, A.; Kadir, M.A.; Islam, K.R.; Islam, K.F.; Mazhar, R.; Hamid, T.; Islam, M.T.; Kashem, S.; Mahbub, Z.B.; et al. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *IEEE Access* **2020**, *8*, 191586–191601. [CrossRef]
46. Qiblawey, Y.; Tahir, A.; Chowdhury, M.E.; Khandakar, A.; Kiranyaz, S.; Rahman, T.; Ibtehaz, N.; Mahmud, S.; Maadeed, S.A.; Musharavati, F.; et al. Detection and severity classification of COVID-19 in CT images using deep learning. *Diagnostics* **2021**, *11*, 893. [CrossRef]
47. Khan, M.M.; Chowdhury, M.E.H.; Arefin, A.S.M.S.; Podder, K.K.; Hossain, M.S.A.; Alqahtani, A.; Murugappan, M.; Khandakar, A.; Mushtak, A.; Nahiduzzaman, M. A Deep Learning-Based Automatic Segmentation and 3D Visualization Technique for Intracranial Hemorrhage Detection Using Computed Tomography Images. *Diagnostics* **2023**, *13*, 2537. [CrossRef]
48. 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 International Publishing: Cham, Switzerland, 2015; pp. 234–241. [CrossRef]
49. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA ML-CDS)*; Springer: Cham, Switzerland, 2018. [CrossRef]
50. Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-Supervised Nets. In Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA; 9–12 May 2015; pp. 562–570.
51. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
52. 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.
53. Wang, S.; Zhang, Y. DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification. *ACM Trans. Multimed. Comput. Commun. Appl.* **2020**, *16*, 3341095. [CrossRef]
54. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In *Artificial Neural Networks and Machine Learning—ICANN 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 270–279.
55. 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; pp. 770–778. [CrossRef]
56. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. February. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; IEEE: Piscataway, NJ, USA, 2017.
57. Jadon, S. A Survey of Loss Functions for Semantic Segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020; pp. 1–7. [CrossRef]
58. Yi-de, M.; Qing, L.; Zhi-Bai, Q. Automated Image Segmentation Using Improved PCNN Model Based on Cross-Entropy. In Proceedings of the 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, China, 20–22 October 2004; pp. 743–746. [CrossRef]
59. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Jorge Cardoso, M. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Cham, Switzerland, 2017; pp. 240–248. [CrossRef]
60. Diederik, P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
61. Ketkar, N. Stochastic Gradient Descent. In *Deep Learning with Python*; Apress: New York, NY, USA, 2017; pp. 113–132.
62. Ravandi, B.; Papapanagiotou, I. A Self-Learning Scheduling in Cloud Software Defined Block Storage. In Proceedings of the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA, 25–30 June 2017; pp. 415–422. [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

A Real-Time Subway Driver Action Sensoring and Detection Based on Lightweight ShuffleNetV2 Network

Xing Shen [1,2] and Xiukun Wei [1,2,*]

[1] School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
[2] State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: xkwei@bjtu.edu.cn

Abstract: The driving operations of the subway system are of great significance in ensuring the safety of trains. There are several hand actions defined in the driving instructions that the driver must strictly execute while operating the train. The actions directly indicate whether equipment is normally operating. Therefore, it is important to automatically sense the region of the driver and detect the actions of the driver from surveillance cameras to determine whether they are carrying out the corresponding actions correctly or not. In this paper, a lightweight two-stage model for subway driver action sensoring and detection is proposed, consisting of a driver detection network to sense the region of the driver and an action recognition network to recognize the category of an action. The driver detection network adopts the pretrained MobileNetV2-SSDLite. The action recognition network employs an improved ShuffleNetV2, which incorporates a spatial enhanced module (SEM), improved shuffle units (ISUs), and shuffle attention modules (SAMs). SEM is used to enhance the feature maps after convolutional downsampling. ISU introduces a new branch to expand the receptive field of the network. SAM enables the model to focus on important channels and key spatial locations. Experimental results show that the proposed model outperforms 3D MobileNetV1, 3D MobileNetV3, SlowFast, SlowOnly, and SE-STAD models. Furthermore, a subway driver action sensoring and detection system based on a surveillance camera is built, which is composed of a video-reading module, main operation module, and result-displaying module. The system can perform action sensoring and detection from surveillance cameras directly. According to the runtime analysis, the system meets the requirements for real-time detection.

Keywords: action recognition; deep learning; driver detection; railway; action sensoring and detection

Citation: Shen, X.; Wei, X. A Real-Time Subway Driver Action Sensoring and Detection Based on Lightweight ShuffleNetV2 Network. *Sensors* **2023**, *23*, 9503. https://doi.org/10.3390/s23239503

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 19 October 2023
Revised: 23 November 2023
Accepted: 27 November 2023
Published: 29 November 2023

1. Introduction

With the development of computer technology, researchers apply advanced artificial intelligence technology in the field of transportation, such as traffic sign detection in the field of road traffic [1,2], railway surface [3–6] and fastener [7,8] defect detection in the field of rail transit. With the rapid development of urban rail transit, the subway system has become the preferred mode of public transportation, which undoubtedly raises higher requirements for the safety of trains. However, there is limited research on subway driver action sensoring and detection based on surveillance cameras. Subway drivers play a crucial role in the safe operation of trains. They need to confirm each step to ensure that no step is missed. The actions of drivers indicate that the current equipment is normally operating. Currently, the monitoring of driver actions is mainly carried out by two surveillance cameras installed in the driver cab. The manual inspection of surveillance videos is used to determine whether the driver has performed the corresponding actions. This method is both inefficient and costly. Therefore, it is of great significance to conduct research on the action sensoring and detection of subway drivers based on surveillance cameras to realize the real-time automatic detection of driver action categories. It can help reduce costs, enhance the operational safety of trains, and improve the intelligence level of urban rail transit monitoring systems.

Subway driver action sensoring and detection (SDASD) belongs to spatial–temporal action detection (STAD), which aims to detect the spatial positions of individuals in the current frame and determine their action categories. In the past, complex handcrafted features have been used, such as spatial–temporal interest points, motion trajectories, etc., for video action recognition [9–13]. These methods achieved good results for simple actions. However, due to the complexity of designing and computing handcrafted features, these methods suffer from slow recognition speed and are not suitable for practical applications.

In recent years, with the rapid development of deep learning, scholars have started to utilize deep neural networks for STAD, which can be categorized into two-stage methods and one-stage methods. The mainstream approach is the two-stage method, where the first stage uses pre-trained object detectors [14–16] to generate human region proposals in the current frame. In the second stage, an action recognition network, often utilizing 3D CNNs to extract spatial–temporal features from video clips, is used for action recognition [17]. Gu et al. [18] introduce an AVA dataset and propose a STAD approach. The region proposal network adopts ResNet50, and the action recognition network uses the I3D network [19], which integrates RGB and optical flow, and finally performs action classification. Subsequently, several models are proposed to improve the performance of STAD, such as ACRN (Actor-Centric Relation Network) [20], STEP (Spatio-TEmporal Progressive) [21], LFB (Long-term Feature Banks) [22], SlowFast [23], Context-Aware RCNN [24], ACARN (Actor–Context–Actor Relation Network) [25] and so on. These two-stage models achieve high accuracy, but the human region detection and action recognition are independent, making these models inefficient. In recent studies, several one-stage models are proposed, where a single network is used for both human region detection and action recognition. These models include WOO (Watch Only Once) [26], SE-STAD (Simple and Efficient Spatial–Temporal Action Detector) [27], and DOAD (Decoupled one-stage action detection network) [28]. However, one-stage models face challenges, such as unsatisfied precision. Deep learning-based algorithms have significantly improved the accuracy of STAD, but the deep convolution network models often have a large number of parameters and computation cost, leading to a slow detection speed that does not meet the real-time detection requirements.

To enable deep neural networks to run on devices with limited computational resources, some lightweight networks have been proposed, such as MobileNet [29–31] and ShuffleNet [32,33]. ShuffleNetV2 [33] reduces the number of parameters and model size significantly by introducing depthwise separable convolutions. It incorporates channel split and channel shuffle to facilitate information exchange between channels. ShuffleNetV2 [33] strikes a good balance between speed and precision.

As of now, there are no reported studies specifically focusing on SDASD based on surveillance cameras. Some researchers have conducted studies on driver action recognition. For instance, Hu et al. [34] propose the RepC3D model, which combines C3D [35] and RepVGG [36] for recognizing subway driver actions. Suo et al. [37] introduce an improved dense trajectory algorithm for driver action recognition. These studies primarily focus on video-level action recognition, where video clips are used for action classification, without explicitly detecting the region of drivers. Different from the above works, our main objective is to apply advanced artificial intelligence technology in the field of subway driver action detection, which is less relevant research work at present. We propose appropriate improvements on the basis of the existing model to further improve the detection performance. And then based on the improved model, we build a real-time driver action detection system to realize real-time video reading from the surveillance camera to carry out action detection, which makes it possible to deploy the system in the subway cab in the future.

This paper aims to achieve real-time sensing of the region of the subway driver and recognition of their action category based on surveillance cameras. For this purpose, a two-stage model for subway driver action sensoring and detection is proposed. In the first stage, the region of the driver is localized by employing a pre-trained lightweight network called MobileNetV2-SSDLite. The network generates driver candidate region proposals along with confidence scores, which are used for subsequent action recognition. In the second stage, an improved ShuffleNetV2 is proposed to extract the spatial–temporal features of the video clips and recognize the category of actions. To boost the performance of network, a spatial enhanced module is introduced to compensate for spatial information loss caused by downsampling. A new branch with larger convolutional kernels is added to expand the receptive field of the network and a shuffle attention module is used to help the network focus the attention on important channels and spatial positions. Experimental results show that the proposed model outperforms other models, achieving a mAP of 72.44%, 4.87% higher than the baseline. Furthermore, a subway driver action sensoring and detection system based on surveillance cameras is built, which performs real-time action detection directly by reading video from surveillance cameras. It is composed of a video-reading module, main operation module and result-displaying module. The performance of the system shows that it meets the requirements for real-time detection. The main contributions are summarized as follows:

1. A real-time subway driver action sensoring and detection model is proposed, which consists of a driver detection network and an action recognition network. The driver detection network is used to locate the region of the driver in the images, and the action recognition network is employed to recognize the category of the action.
2. A spatial enhanced module is introduced after the first convolution downsampling layer, compensating for the loss of spatial information and enhancing the spatial positions of the feature map. In addition, a dataset specifically for subway driver action sensoring and detection is constructed.
3. A new branch with a large convolutional kernel in the shuffle units is proposed to expand the receptive field, which is crucial for the subsequent action recognition. In addition, the shuffle attention module is introduced to help the network focus the attention on important channels and spatial positions.
4. A real-time subway driver action sensoring and detection system based on surveillance cameras is built, which reads video from surveillance cameras and performs SDASD directly. According to the runtime analysis, the system meets the requirements for real-time detection.

The rest of this paper is organized as follows. Section 2 introduces the problems studied in this paper. In Section 3, a lightweight two-stage model for subway driver action sensoring and detection is proposed. Section 4 introduces the detailed experiments and results. In Section 5, a subway driver action sensoring and detection system is introduced. Section 6 summarizes the conclusions and presents an outlook for future work.

2. Problem Statements

Subway drivers play a crucial role in ensuring the safe operation of urban rail transit. They need to confirm each step with their fingers to ensure that no step is missed. The actions of the driver can be used to determine whether the equipment is normally operating or not. Currently, the monitoring of driver actions and their states is primarily performed through the installation of two surveillance cameras in the driver cab (one located at the bottom left corner and the other at the top right corner). The action monitoring uses manual inspection of the recorded videos to check the actions of driver. This manual approach is inefficient and costly.

With the trend of intelligent development in urban rail transit, more and more advanced artificial intelligence algorithms are being applied to detection tasks. However, there is limited research on monitoring subway driver actions. Existing studies mostly focus on classifying driver actions from 2D images, namely recognizing the action category in a single image. Such methods can only identify simple actions without temporal relationships and do not achieve comprehensive action recognition. An action is often composed of multiple consecutive frames with temporal dependencies, and relying on a single image is insufficient for recognizing actions with temporal relationships. Suo et al. [37] study subway driver action recognition by video clips from the perspective of the video level. The actions can be categorized as arrival confirmation, departure confirmation, interval confirmation, platform closing confirmation, and no action. An improved dense trajectory-based method has been proposed for recognizing driver gesture actions. While this method achieves high accuracy, it suffers from slow speed and is used for pre-cropped action video clips; thus, it cannot provide the real-time localization of the driver region and the recognition of driver actions.

In this paper, the action sensoring and detection of subway drivers based on surveillance cameras is studied, aiming to sense the region of the driver and recognize their current actions from surveillance videos. According to surveillance videos in subway driver cabs, driver actions and states are categorized into 11 classes as shown in Figure 1, including sitting (Sit), standing inside the cab (StinCab), standing outside the cab (StoutCab), walking from inside to outside (WafrI2O), walking from outside to inside (WafrO2I), pointing to the instrument and screen (Po2InSc), pointing to the front window (Po2FrWin), pointing to the lower left instrument (Po2LLin), pressing the door button (PrDoBu), pushing the instrument (PuIn), and no action (None). For the task of driver region localization, the lightweight object detection algorithm MobileNetv2-SSDLite is employed to locate the region of the driver in the image, and the coordinates and confidence scores are obtained. These coordinates and scores are used as regions of interest (ROIs) and fed to the action recognition network. For driver action recognition, an improved lightweight ShuffleNetv2 is proposed to extract spatial–temporal features from multiple frames of input. The ROIs from the driver region localization task are mapped onto the final feature map, followed by ROI pooling to generate fixed-size features for ROIs, and then processed through fully connected layers for action recognition. To evaluate the real-time performance of the proposed model, a subway driver action sensoring and detection system based on surveillance cameras is built, consisting of three parts: video-reading module, main operation module, and result-displaying module. The video-reading module stores the video streams in a reading queue, the main operation module samples frames from the queue and performs driver region localization and driver action recognition on the sampled frames, and the result-displaying module renders the results on the frames and composes the video for display on the screen. A surveillance camera is installed, and the system can perform action sensoring and detection on the video streams from the surveillance camera. According to the runtime of each module, the system achieves real-time detection.

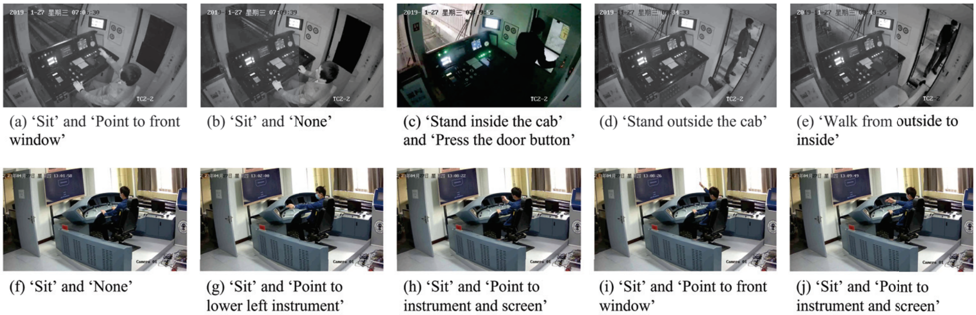

(a) 'Sit' and 'Point to front window'
(b) 'Sit' and 'None'
(c) 'Stand inside the cab' and 'Press the door button'
(d) 'Stand outside the cab'
(e) 'Walk from outside to inside'
(f) 'Sit' and 'None'
(g) 'Sit' and 'Point to lower left instrument'
(h) 'Sit' and 'Point to instrument and screen'
(i) 'Sit' and 'Point to front window'
(j) 'Sit' and 'Point to instrument and screen'

Figure 1. Action categories. The images in the first rows are from the surveillance video of the driver cab, and the images in the second rows are from simulated video.

3. Methodology

3.1. Overall Framework

The overall framework of the model is shown in Figure 2. The model is composed of the driver detection network and the action recognition network. The driver detection network generates proposals of the driver region, and the action recognition network recognizes the current action category of the driver. To achieve fast and accurate detection, lightweight networks are employed for both tasks. For the driver detection network, a pre-trained MobileNetV2-SSDLite [30] is utilized, which has a small number of parameters and model size, allowing it to quickly and accurately detect the region of the driver. For the action recognition network, an improved ShuffleNetV2 is proposed. The network also has a small number of parameters and model size, enabling the fast and accurate recognition of actions.

3.2. Driver Detection Network

In this paper, the pre-trained MobileNetv2-SSDLite is adopted for the driver detection network. It aims to sense and locate the region of the driver in the image and obtain the coordinates and confidence scores. The SSD [16] is a classic one-stage object detection algorithm that can simultaneously perform object localization and classification in a single stage. It combines the advantages of the anchor-based mechanism from region proposal algorithms and the regression-based algorithm in one-stage methods, resulting in high accuracy and fast detection speed. The original SSD uses VGG16 as a base network. However, the large number of parameters in the VGG16 makes it unsuitable for running on resource-limited embedded devices and mobile devices. To address this issue, the SSDLite object detection network based on MobileNetV2 is proposed, which reduces the number of parameters and computation. Specifically, the VGG16 is replaced with MobileNetV2 for feature extraction. Additionally, instead of using regular convolutions, the extra convolution layers in SSDLite utilize depthwise separable convolutions (DWConv) as the basic structure. The input size of image is 320 × 320. Predictions are made by using six different-sized feature maps when detecting, and proposals of the driver region along with confidence scores are obtained. These proposals are then selected by non-maximum suppression (NMS) to obtain the final proposals, used for subsequent action recognition.

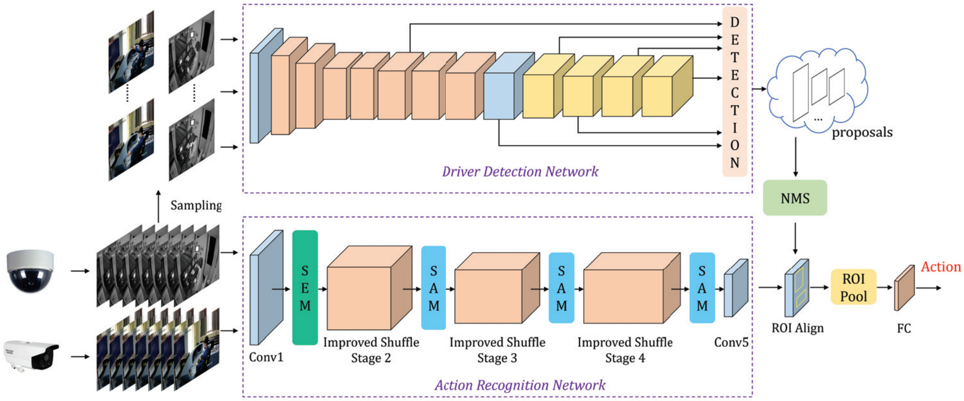


Figure 2. The overall framework of the model.

3.3. Action Recognition Network

3.3.1. ShuffleNetV2

Three-dimensional ShuffleNetV2 [33] is a lightweight network with small number of parameters and computation. The shuffle units of 3D ShuffleNetV2 are shown in Figure 3. When the stride is one, the channels are split into two branches. Branch 1 does not have any operations, while branch 2 adopts DWConv. The outputs of branch 1 and branch 2 are concatenated, followed by a channel shuffle module. When the stride is two, the input is processed by two branches. The outputs of branch 1 and branch 2 are concatenated. The channel shuffle operation allows for information exchange between channels. The channel shuffle module is shown in Figure 4. For a given feature map with a specific number of channels, the channels are first divided into G groups. Then, the groups are transposed and rearranged to obtain the shuffled feature map. The channel shuffle operation does not introduce any parameters and achieves channel-wise information exchange through simple grouping and transposition operations.

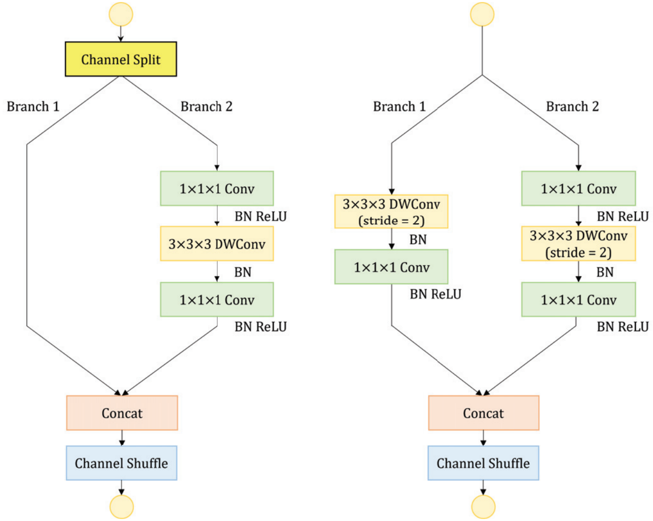


Figure 3. The structure of the shuffle units.

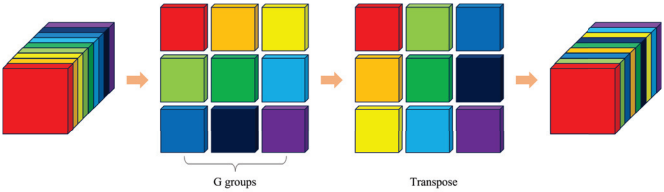


Figure 4. The structure of the channel shuffle.

3.3.2. Spatial Enhanced Module (SEM)

When the network performs convolutional downsampling, the size of the feature maps decreases. Though this allows for capturing high-level features, downsampling leads to spatial information loss. To reduce the information loss, a SEM is added after the first convolution to enhance the spatial representation capability of network. The specific structure is shown in Figure 5. Firstly, the global average pooling and global max pooling are adopted. The resulting feature maps are then concatenated and processed by a 3D convolution to extract features. An activation function is applied to enhance the representation ability of the network, and the feature maps are multiplied element-wise with the original feature maps to obtain the enhanced feature maps. The formulas are denoted as follows :

$$M_{SA} = \sigma(conv([AvgPool(X); MaxPool(X)])) \tag{1}$$

$$X^{'} = X \otimes M_{SA} \tag{2}$$

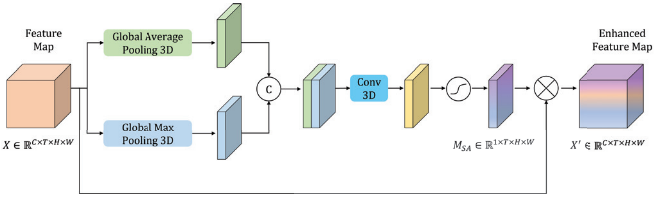

Feature Map
$X \in \mathbb{R}^{C \times T \times H \times W}$
Global Average Pooling 3D
Global Max Pooling 3D
C
Conv 3D
$M_{SA} \in \mathbb{R}^{1 \times T \times H \times W}$
Enhanced Feature Map
$X' \in \mathbb{R}^{C \times T \times H \times W}$

Figure 5. The structure of the spatial enhanced module.

3.3.3. Improved Shuffle Units (ISUs)

The receptive field of the network can be expanded by using large convolution kernels, which is crucial for subsequent tasks. Inspired by it, a new branch with a 5 × 5 × 5 kernel size is added to the ShuffleNet units as shown in Figure 6. When the stride is 1, branch 3 is added to obtain a larger receptive field. The rest of branch 3 is the same as branch 2. To ensure that the final concatenation has the same number of channels, the output channels of the last convolution in branch 2 and branch 3 are set to 1/4 of the input channels. Branch 1, branch 2, and branch 3 are then concatenated to obtain feature maps, followed by a channel shuffle module. When the stride is 2, branch 3 and branch 4 are added with a bigger kernel size. The output channels of the last convolution in all branches are set to 1/4 of the input channels. Branch 1, branch 2, branch 3, and branch 4 are then concatenated to obtain the final feature maps, followed by a channel shuffle module. The introduction of improved shuffle units increases the number of computations, about 1.5 times to 2 times as much as the original shuffle units. The detailed calculation procedure is shown in Appendix A.

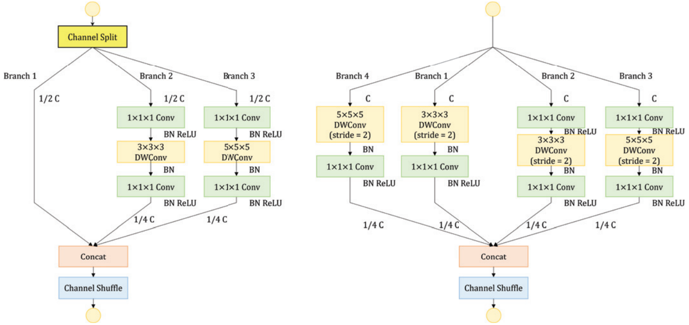


Figure 6. The structure of the improved shuffle units.

3.3.4. Shuffle Attention Module (SAM)

The attention module allows the network to focus on important features and suppress unimportant features. There are two main types of attention mechanisms: channel attention and spatial attention. Channel attention focuses on “what”, while spatial attention focuses on “where”. In this paper, to boost the representation ability in both the spatial and channel dimensions of the feature maps, the shuffle attention module (SAM) [38] is added after each improved shuffle stage. The structure of SAM is shown in Figure 7. First, the feature maps are divided into g groups along the channel dimension. Each group is further divided into two branches, namely, the channel attention branch and the spatial attention branch. These branches are responsible for generating different channel and spatial importance weights.

Channel attention branch: Instead of using the traditional SE [39] module, which introduces a large number of parameters, a simple combination of global average pooling, scale, and sigmoid is adopted. Firstly, global average pooling is used to embed global information, generating $s \in \mathbb{R}^{\frac{c}{2g} \times 1 \times 1 \times 1}$.

$$s = F_{gp}(X_{k1}) = \frac{1}{T \times H \times W} \sum_{i=1}^{T} \sum_{j=1}^{H} \sum_{k=1}^{W} X_{k1}(i,j,k) \tag{3}$$

$$X_{k1}^{'} = \sigma(F_c(s)) \cdot X_{k1} = \sigma(W_1 s + b_1) \cdot X_{k1} \quad (4)$$

Spatial attention branch: Unlike the channel attention branch, spatial attention focuses on the spatial dimension. Firstly, Group Norm (GN) is used to obtain statistical information along the spatial dimensions. Then, Fc(.) is applied to enhance the spatial attention. The formula is denoted as follows:

$$X_{k2}^{'} = \sigma(W_2 \cdot GN(X_{k2}) + b_2) \cdot X_{k2} \quad (5)$$

Aggregation: After obtaining the channel attention weights and spatial attention weights, it is necessary to aggregate them. Firstly, a simple concatenation operation is adopted. Then, inter-group information exchange is performed by channel shuffle module.

The computation of the grouping operation in SAM is about $\frac{1}{4g^2}$ of that of the non-grouping operation. The detailed calculation procedure is shown in Appendix B.

3.3.5. Network Structure

After a detailed introduction of each component of the network, the complete action recognition network is presented along with its structure and specific parameters in Table 1. The network consists of two standard convolutions, one max pooling, one global average pooling, one SEM module, three improved shuffle stages, three SAM modules, one ROI Align and Pooling, and one fully connected layer. In Table 1, T represents the number of input frames, and there is no downsampling in the temporal dimension.

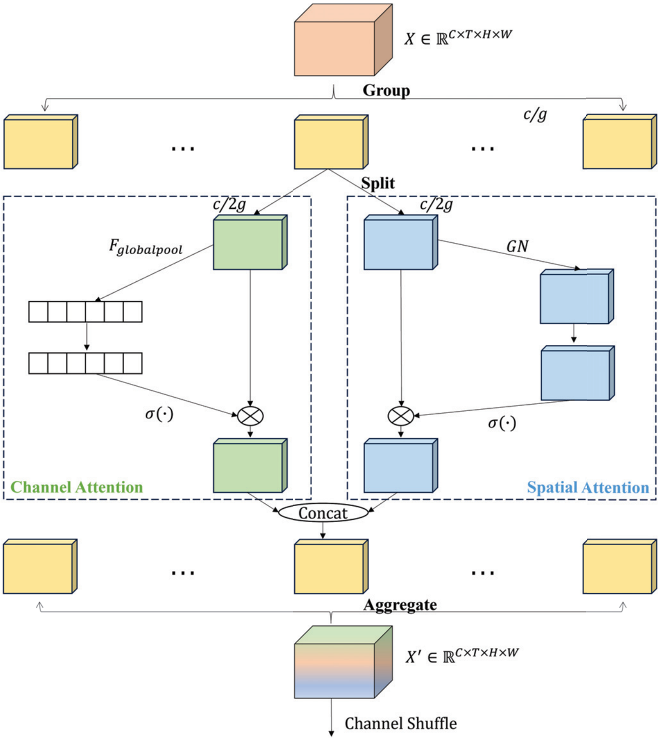


Figure 7. The structure of the shuffle attention module.

Table 1. Structure and parameters of the action recognition network.

Operator	Output Size	KSize	Stride	Rep.	Output Channels
Frames	T × 256 × 256	-	-	-	3
Conv1	T × 128 × 128	3 × 3 × 3	(1,2,2)	1	24
MaxPool	T × 64 × 64	3 × 3 × 3	(1,2,2)	1	24
SEM	T × 64 × 64	-	-	-	24
Improved Shuffle Stage2	T × 32 × 32	-	(1,2,2), (1,1,1)	1, 3	116
SAM	T × 32 × 32	-	-	-	116
Improved Shuffle Stage3	T × 16 × 16	-	(1,2,2), (1,1,1)	1, 7	232
SAM	T × 16 × 16	-	-	-	232
Improved Shuffle Stage4	T × 16 × 16	-	(1,1,1)	4	464
SAM	T × 16 × 16	-	-	-	464
Conv5	T × 16 × 16	1 × 1 × 1	(1,1,1)	1	1024
ROI Align & Pool	1 × 8 × 8	-	-	-	1024
Global Average Pool	1 × 1 × 1	-	-	-	1024
Fully Connected	-	-	-	-	12

KSize represents kernel size, Rep. represents repeat number.

4. Experiments

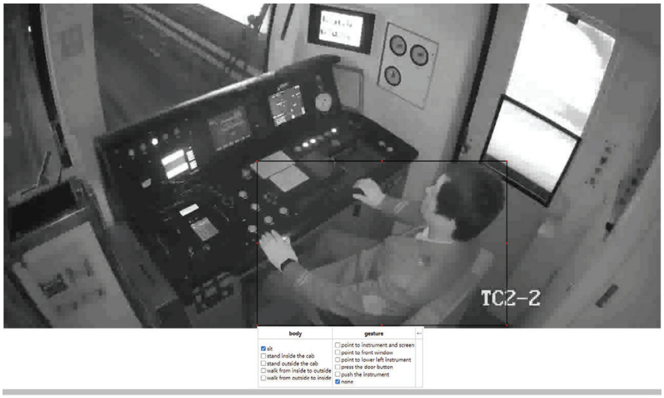
4.1. Dataset Preparation

The videos of the driver cab used in the experiment are from Beijing Metro Line 9, with a video resolution of 1280 × 720. The simulation video is recorded by the surveillance camera installed in the lab, with a video resolution of 1920 × 1080. Each raw surveillance video is about an hour long, and if it is fed directly into the model, it requires a huge amount of memory and computing resources. Therefore, in order to better sense and detect the category of the subway driver action, 328 video clips are cropped with a duration of 10 s from the original driver cab videos and simulation videos, of which 163 clips are from the actual surveillance video and 165 clips are from laboratory simulation videos. Each clip contains two action labels. The action labels includes sitting (Sit), standing inside the cab (StinCab), standing outside the cab (StoutCab), walking from inside to outside (WafrI2O), walking from outside to inside (WafrO2I), pointing to the instrument and screen (Po2InSc), pointing to the front window (Po2FrWin), pointing to the lower left instrument (Po2LLin), pressing the door button (PrDoBu), pushing the instrument (PuIn), and no action (None). The number of labeled actions is shown in Table 2. The driver detection network does not require additional datasets. As the proposed model works, the frames for the driver detection network are sampled from the input video clip and no additional datasets are required.

The annotation method refers to the AVA dataset format [18]. First, the video is extracted into a series of frames with an FPS of 30. The AVA dataset format does not label all frames but annotates 1 frame per second. Therefore, in the spatial–temporal detection dataset of the subway driver action, the first frame per second is annotated [40]. Since the first and last 2 s of the videos are not involved in detecting, only images with indexes of 61, 91, 121, 151, 181, 211, 241 are labeled for each video clip with a duration of 10 s. In order to quickly label the region of the driver, the pre-trained YOLOv5 [41] is used to detect the region of the driver, and the coordinates and confidence scores of the proposals are obtained as rough labeling, and then the rough labeling is imported into the VIA [42] labeling tool for action category labeling. The annotation process is shown in Figure 8.

Table 2. The number of labeled actions.

Action	Training Set	Testing Set	Total
Sit	953	402	1355
StinCab	157	89	246
StoutCab	326	147	473
WafrI2O	82	25	107
WafrO2I	91	23	114
Po2InSc	109	31	140
Po2FrWin	62	21	83
Po2LLin	61	22	83
PrDoBu	98	58	156
PuIn	268	150	418
None	465	181	646
Total	2672	1149	3821



body	gesture
☑ sit	□ point to instrument and screen
□ stand inside the cab	□ point to front window
□ stand outside the cab	□ point to lower left instrument
□ walk from inside to outside	□ press the door button
□ walk from outside to inside	□ push the instrument
	☑ none

Figure 8. The annotation process.

The detailed process by which MobileNetV2-SSDLite results are used by improved shuffleNetV2 to drive the action classification task is as follows.

Taking a cropped video clip for example, the first and last 2 s of the videos are not involved in detecting (AVA dataset format), and the index of 61, 91, 121, 151, 181, 211, and 241 frames are labeled. Therefore, taking these seven frames as the center, and sampling eight frames each, we can obtain seven clips for the model training, where each clip has eight frames.

Taking the frame with index 61 as an example, with 61 as the center and an interval of 8, a total of 8 frames are sampled, that is, the index corresponding to the sampled frames is (29, 37, 45, 53, 61, 69, 77, 85), and the action label is the same as the label of the frame with index 61. For the driver detection network, these eight frames are input in sequence to obtain the driver proposals (namely anchors) in the eight frames. Then, the non-maximum suppression (NMS) is then used to filter invalid anchors that exceed a fixed threshold. The retained anchors after NMS are then mapped on the feature map of the last convolution layer of the improved ShuffleNetV2 (namely the output feature map of conv5, as shown in Figure 1). For the action recognition network, these eight frames are taken as an input. The region of the driver on the feature map, namely the region of interest (ROI), can be obtained, and then the ROIs which have different sizes are transformed into fixed sizes by ROI pooling, and finally the fixed size features are sent to the fully connected layer for action classification.

4.2. Evaluation Indicators and Experimental Details

The purpose of subway driver action sensoring and detection is to sense and locate the region of the driver and recognize the category of the action, paying more attention to driver action recognition. A common evaluation indicator is mAP (mean Average Precision), which is the average of AP in all action categories.

In this paper, all the models are implemented by PyTorch and trained on 1 NVIDIA A6000 GPU. The CPU is Inter(R) Xeon(R) silver 4314 @2.4 GHz.

The input of the model is eight frames, which are sampled at equal intervals. The interval in this paper is set to eight, that is, one frame is sampled every eight frames. When model training, eight images are scaled to 256 × 256 after sampling, and horizontal flip is introduced to augment the dataset. For model testing, the height of the image is scaled to 256, and the width is scaled proportionally. The parameters are set as follows: the optimizer is SGD, the initial learning rate is 0.01, the weight momentum is 0.00003, and the learning rate decay strategy is cosine annealing, where warmup_iters is 500, warmup_ratio is 0.1, the minimum learning rate is 0.00001, and the training epoch is 200.

4.3. Experimental Results

For the driver detection network, the Intersection over Union (IoU) threshold is set to 0.5. For the action recognition network, the prediction score threshold is set to 0.9. The experimental results are shown in Table 3. The baseline is the original 3D ShuffleNetV2 [33]. As can be seen from Table 3, SEM, ISU, and SAM can improve the performance of the model. ISU brings the most obvious gain; the mAP can increase 4.13%, and the number of parameters is smaller than the baseline. When all modules are added to the network, the mAP increases to 72.44%, 4.87% higher than the baseline.

Table 3. Ablation experiments.

Baseline	SEM	ISU	SAM	mAP
✓				0.6757
✓	✓			0.6842
✓		✓		0.7170
✓			✓	0.7114
✓	✓	✓	✓	**0.7244**

Table 4 shows the AP of each action. It can be seen that the introduction of SEM, ISU and SAM improve the AP of most action categories, among which walking from inside to outside has the most obvious improvement, from 0.5143 to 0.8309, improved by 31.66%, followed by pressing the door button, which increases from 0.2172 to 0.3276, an improvement of 11.04%. Though the AP of walking from outside to inside, pointing to the instrument and screen, and pushing the instrument decrease slightly, other categories of actions have different amplitudes of increment. It can be concluded that the introduction of SEM, ISU and SAM improve the performance of the model. In addition, it is seen that the AP of pressing the door button action is low, owing to the small number of action instances in the actual surveillance video, and this action is not simulated due to the constraints of the operation console, which further leads to a smaller number of actions than other actions, resulting in the model not learning the feature of action well. Figure 9 shows the confusion matrix for the recognition results of the proposed model. The action recognition network can almost accurately classify all kinds of actions; only seven actions are classified incorrectly. Of these seven actions, one StinCab and one StoutCab are classified as Sit because the driver is close to the seat in the two video clips, leading to the wrong classification. One StinCab is classified as WafrI2O because the StinCab occurs at the junction where the two actions occur, resulting in a classification error. Two WafrO2I are classified as StoutCab because these actions are both outside the cab and the action after StoutCab is WafrO2I. One WafrI2O is classified as WafrO2I because at the door, WafrI2O is similar to WafrO2I. One WafrO2I is classified as PuIn because in the real cab video, the action behind WaFrO2I is PuIn, resulting in a recognition error at the action connection. Overall, it can be seen that the proposed model can well recognize the action categories of the driver.

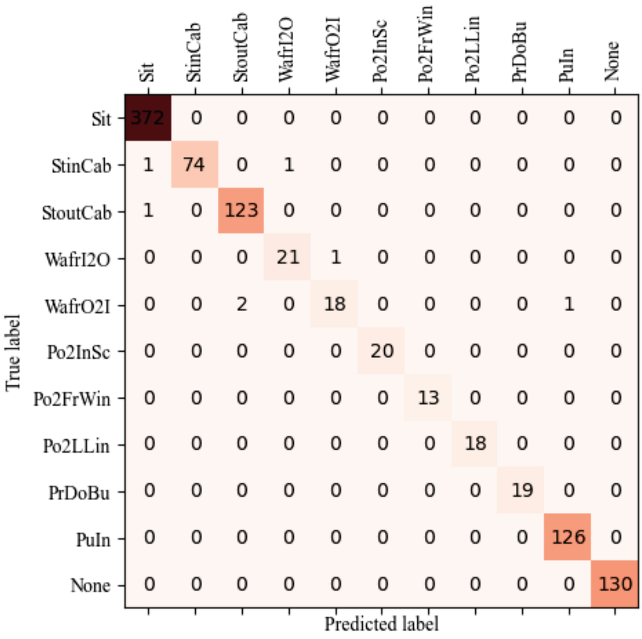
Table 4. The AP of actions.

Action	Baseline	+SEM	+ISU	+SAM	+SEM+ISU+SAM
Sit	0.8760	0.8672	0.8804	0.8680	**0.8817**
StinCab	0.7865	0.8090	0.8315	0.8090	**0.8315**
StoutCab	0.8231	**0.8503**	0.8360	0.8298	0.8296
WafrI2O	0.5143	0.8696	**0.8783**	0.8200	0.8309
WafrO2I	0.8239	0.7826	**0.8261**	0.7712	0.7666
Po2InSc	0.5839	0.4073	**0.6144**	0.4315	0.5823
Po2FrWin	0.5714	0.6190	0.5159	0.6667	**0.6122**
Po2LLin	0.7727	0.7246	0.7702	0.7727	**0.8182**
PrDoBu	0.2172	0.1724	0.2165	0.3575	**0.3276**
PuIn	0.7967	0.7463	**0.8240**	0.8047	0.7865
None	0.6670	0.6774	0.6936	0.6940	**0.7013**
mAP	0.6757	0.6842	0.7170	0.7114	**0.7244** (+4.87%)

In order to prove that the SAM module is better than the SE module, we carried out a comparative experiment. Table 5 shows the comparison results. It can see that the SAM has fewer parameters and better performance than the SE module.

Table 5. Comparison between SE and SAM.

Model	Model Size	Parameters	GFlops	mAP
+SEM+ISU+SE	10.50 M	1,329,416	6.36	0.7230
+SEM+ISU+SAM	10.26 M	1,294,740	6.36	0.7244



	Sit	StinCab	StoutCab	WafrI2O	WafrO2I	Po2InSc	Po2FrWin	Po2LLin	PrDoBu	PuIn	None
Sit	372	0	0	0	0	0	0	0	0	0	0
StinCab	1	74	0	1	0	0	0	0	0	0	0
StoutCab	1	0	123	0	0	0	0	0	0	0	0
WafrI2O	0	0	0	21	1	0	0	0	0	0	0
WafrO2I	0	0	2	0	18	0	0	0	0	1	0
Po2InSc	0	0	0	0	0	20	0	0	0	0	0
Po2FrWin	0	0	0	0	0	0	13	0	0	0	0
Po2LLin	0	0	0	0	0	0	0	18	0	0	0
PrDoBu	0	0	0	0	0	0	0	0	19	0	0
PuIn	0	0	0	0	0	0	0	0	0	126	0
None	0	0	0	0	0	0	0	0	0	0	130

Figure 9. The confusion matrix of improved ShuffleNetV2.

In order to evaluate the performance of the proposed network, 3D MobileNetV1 [29], 3D MobileNetV3 [31], SlowFast-R50 [23], SlowOnly-R50 [23], and SE-STAD [27] are selected for comparison. The experimental results are shown in Table 6. The results show that the proposed network has a higher mAP and a smaller number of parameters and model size than the compared networks. Table 7 shows the AP of each action. It can be seen that the APs of the proposed model are better than the compared models in most action categories, and the proposed network reaches the state of the art.

Table 6. Comparison with other networks.

Model	mAP	Parameters	Model Size	GFlops
3D MobileNetV1 [29]	0.6164	3,310,284	25.43 M	14.60
3D MobileNetV3 [31]	0.7189	1,165,964	9.07 M	3.39
SlowFast-R50 [23]	0.6095	33,671,220	257.46 M	193.57
SlowOnly-R50 [23]	0.6630	31,659,084	241.91 M	166.08
SE-STAD [27]	0.7068	40,650,557	310 M	213.65
ours	**0.7244**	1,294,740	10.26 M	6.36

Table 7. The AP of actions.

Action	3D MobileNetV1 [29]	3D MobileNetV3 [31]	SlowFast-R50 [23]	SlowOnly-R50 [23]	SE-STAD [27]	Ours
Sit	0.8742	0.8737	0.8753	0.869	0.9801	0.8817
StinCab	0.7978	0.8427	0.7978	0.8539	0.8202	0.8315
StoutCab	0.8298	0.8299	0.8639	0.8502	0.9048	0.8296
WafrI2O	0.76	0.8345	0.64	0.781	0.4	0.8309
WafrO2I	0.5652	0.7826	0.3913	0.6957	0.3478	0.7666
Po2InSc	0.4649	0.5755	0.5066	0.5145	0.7041	0.5823
Po2FrWin	0.1905	0.6444	0.4459	0.4238	0.7963	0.6122
Po2LLin	0.7273	0.7727	0.5455	0.6364	0.8636	0.8182
PrDoBu	0.1552	0.1551	0.1207	0.1207	0.1034	0.3276
PuIn	0.7929	0.8174	0.8266	0.8259	0.9267	0.7865
None	0.6231	0.7789	0.6906	0.7222	0.9282	0.7013
mAP	0.6164	0.7189	0.6095	0.663	0.7068	0.7244

Figure 10 shows the driver action detection results from the actual surveillance video in the driver cab. It can be seen that the model can precisely locate the region of the driver and recognize the category of action.

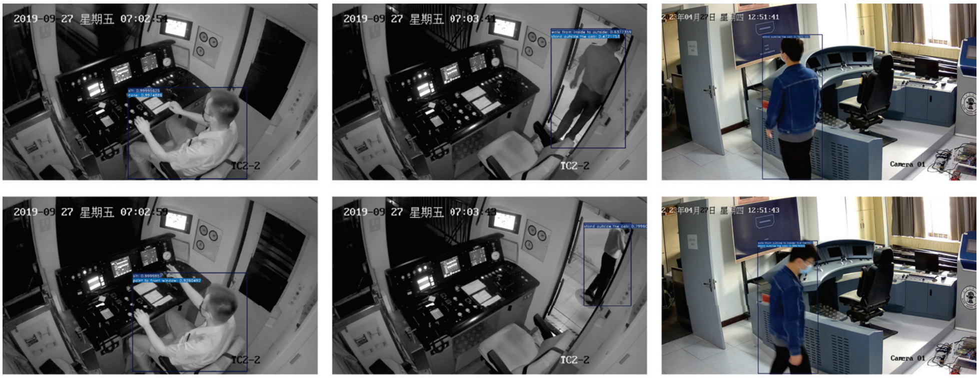

2019-09-27 星期五 07:02:5
TC2-2
2019-09-27 星期五 07:03:4
TC2-2
12:51:41
Camera 01
2019-09-27 星期五 07:02:5
TC2-2
2019-09-27 星期五 07:03:4
TC2-2
12:51:43
Camera 01

Figure 10. *Cont.*

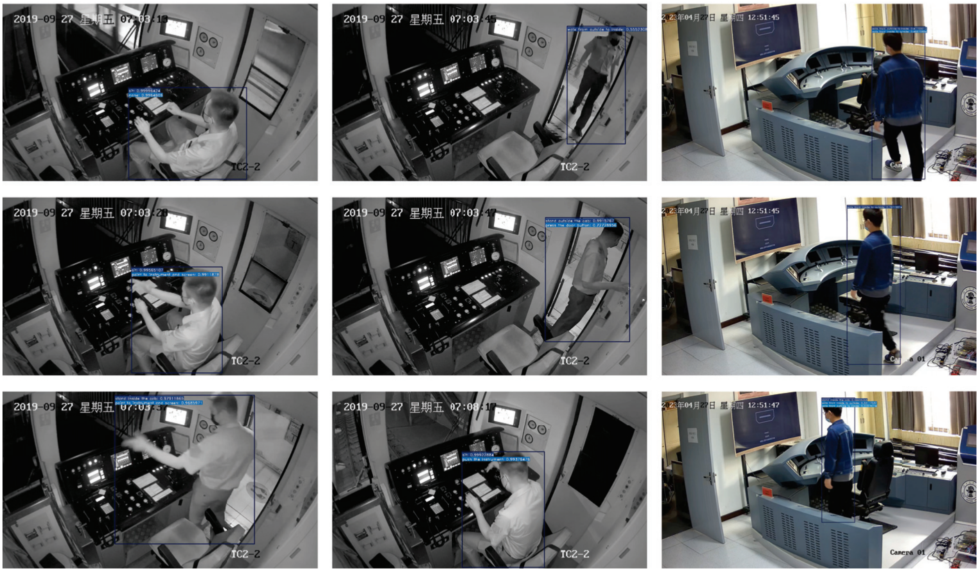


Figure 10. The action sensoring and detection result of the subway driver.

5. Subway Driver Action Sensoring and Detection System

5.1. System Structure

Based on the proposed model, a subway driver action sensoring and detection system (SDASD) is built, aiming to read the video from the surveillance camera in real-time and conduct SDASD directly. The system structure is shown in Figure 11. The system is composed of a video-reading module, main operation module and result-displaying module.

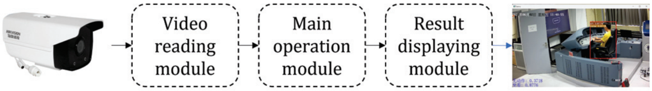


Figure 11. The structure of subway driver action sensoring and detection system.

5.1.1. Video-Reading Module

The real-time video is read from the surveillance camera, and video frames are stored in the reading queue for the main operation module.

5.1.2. Main Operation Module

Frames are sampled from the reading queue according to the given sampling strategy (that is, sampling 1 frame at an interval of 8, and sampling 8 frames in total). The eight frames are sent to the driver detection network to predict the region of the driver, and the coordinates and confidence scores of the proposals are obtained. The region proposals are drawn in the frames, which are taken as the ROIs in the subsequent action recognition. In addition, the eight frames are pre-processed for the action recognition network. In the last layer of convolution, the ROIs in the driver detection network are mapped on the feature map, and finally the ROI pooling is carried out to obtain the fixed-size feature map. Finally, the action categories and corresponding confidence scores are predicted through the fully connected layer.

5.1.3. Result-Displaying Module

The result-displaying module mainly draws the action recognition results on the frames, including the action category and its corresponding confidence score, then synthesizes the video at a fixed frame rate, and finally displays it on the screen.

5.2. Performance Evaluation

To evaluate the performance of the system, experiments are conducted on a personal laptop with a Core (TM) i5-12500H 2.5 GHz CPU and an NVIDIA RTX 2050 (4 GB) GPU. Since the input images are sampled from 64 frames (about 64/30 = 2.13 s), that is, the model needs to complete the whole process (video-reading module, main operation module, and result-displaying module) within 2.13 s to meet the real-time detection requirements. Table 8 shows the time consumed of each module. It can be seen that the complete runtime of our model is between 0.6 s and 0.75 s, which is much less than 2.13 s. Compared with the other models, our model is better regarding total runtime. Therefore, it is concluded that the model meets the requirements of real-time action detection from surveillance cameras. The results of the real-time detection system are shown in Figure 12.

Table 8. The runtime of each module.

Model	Video-Reading Module	Main Operation Module	Result-Displaying Module	Total
3D MobileNetV1 [29]	280–350 ms	120–180 ms	300–350 ms	700–880 ms
3D MobileNetV3 [31]	250–350 ms	100–180 ms	280–350 ms	630–880 ms
SlowFast-R50 [23]	280–350 ms	360–450 ms	380–450 ms	1020–1250 ms
SlowOnly-R50 [23]	280–350 ms	220–280 ms	250–300 ms	750–930 ms
ours	250–300 ms	100–150 ms	250–300 ms	600–750 ms

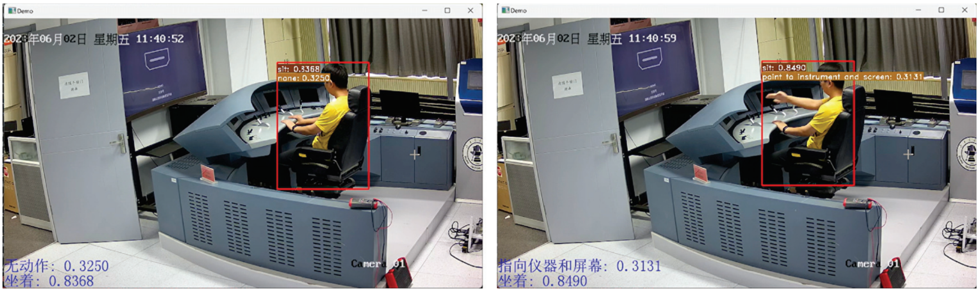


Figure 12. The results of real-time detection system.

6. Conclusions

In this paper, a lightweight two-stage model for subway driver action sensoring and detection based on surveillance cameras is proposed. It consists of the driver detection network and the action recognition network. The driver detection network adopts MobileNetV2-SSDLite, with the purpose of locating the region of the driver. The action recognition network employs the improved ShuffleNetV2 to extract spatial–temporal features and recognizes the category of action. The proposed network has a smaller number of parameters and model size than the compared networks. The experimental results show that the proposed network outperforms the compared networks, with a mAP of 72.44%, 4.87% higher than the baseline. Then a subway driver action sensoring and detection system is built based on the proposed model to realize real-time detection from surveillance cameras. The system runs on a personal laptop; according to the runtime of the system, it takes 0.6 to 0.75 s for a whole process, which is less than the video duration of 2.13 s. It can be seen that the system meets the real-time detection requirements.

In the future research, we will further optimize our system, expand the dataset, and improve the performance. In addition, the action statistics function is taken into account for the subway driver action sensoring and detection to count the number of actions completed by the driver.

Author Contributions: Conceptualization, X.S.; methodology, X.S.; writing—original draft preparation, X.S.; writing—review and editing, X.S. and X.W.; supervision, X.W.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (grant number 2022YFB4301204).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

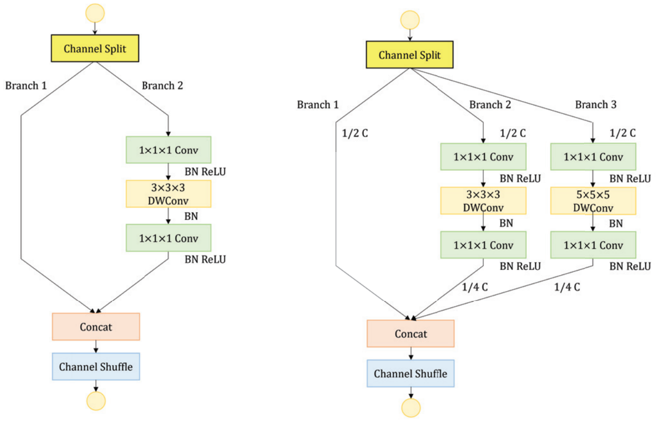
Data Availability Statement: Data are contained within the article.

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

Appendix A

The additional computation resulting from the improved shuffle unit (ISU) is calculated as follows.

When the stride is one, the input shape is $X \in \mathbb{R}^{C\times T\times H\times W}$, and the output shape is $X' \in \mathbb{R}^{C\times T\times H\times W}$. The structures of the original shuffle units and improved shuffle units are shown in Figure A1.



(a) 3D ShuffleNetV2 Unites(stride=1)

(b) Improved 3D ShuffleNetV2 Unites(stride=1)

Figure A1. The structures of the original shuffle units and improved shuffle units (stride = 1).

The computation of 3D ShuffleNetV2 (stride = 1):

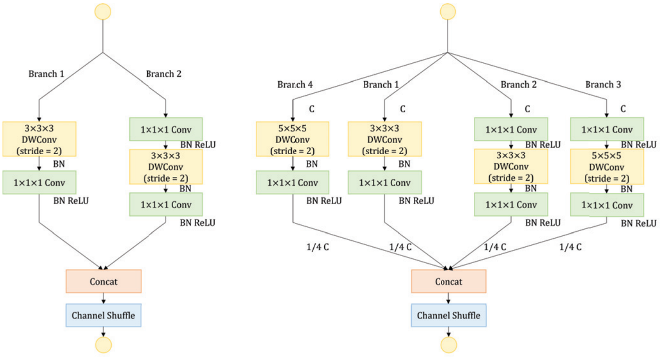
$$\begin{aligned} Comput\_original1 &= (2\times\frac{C}{2}\times 1\times 1\times 1-1)\times\frac{C}{2}\times T\times H\times W \\ &+(\frac{2\times\frac{C}{2}\times 3\times 3\times 3}{\frac{C}{2}}-1)\times\frac{C}{2}\times T\times H\times W \\ &+(2\times\frac{C}{2}\times 1\times 1\times 1-1)\times\frac{C}{2}\times T\times H\times W \\ &=\frac{C}{2}\times T\times H\times W\times(2C+51) \end{aligned} \tag{A1}$$

The computation of improved 3D ShuffleNetV2 (stride = 1):

$$\begin{aligned}
Comput\_improved1 &= (2 \times \frac{C}{2} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2} \times T \times H \times W \\
&+ (\frac{2 \times \frac{C}{2} \times 3 \times 3 \times 3}{\frac{C}{2}} - 1) \times \frac{C}{2} \times T \times H \times W \\
&+ (2 \times \frac{C}{2} \times 1 \times 1 \times 1 - 1) \times \frac{C}{4} \times T \times H \times W \\
&+ (2 \times \frac{C}{2} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2} \times T \times H \times W \\
&+ (\frac{2 \times \frac{C}{2} \times 5 \times 5 \times 5}{\frac{C}{2}} - 1) \times \frac{C}{2} \times T \times H \times W \\
&+ (2 \times \frac{C}{2} \times 1 \times 1 \times 1 - 1) \times \frac{C}{4} \times T \times H \times W \\
&= \frac{C}{2} \times T \times H \times W \times (3C + 299)
\end{aligned} \tag{A2}$$

It can be seen that when the stride is one, the introduction of improved shuffle units increases the number of computations, about 1.5 times as much.

When the stride is two, the input shape is $X \in \mathbb{R}^{C \times T \times H \times W}$, and the output shape is $X' \in \mathbb{R}^{C \times T \times \frac{H}{2} \times \frac{W}{2}}$. The structures of the original shuffle units and improved shuffle units are shown in Figure A2.



(a) 3D ShuffleNetV2 Unites(stride=2)

(b) Improved 3D ShuffleNetV2 Unites(stride=2)

Figure A2. The structures of original shuffle units and improved shuffle units (stride = 2).

The computation of 3D ShuffleNetV2 (stride = 2):

$$\begin{aligned}
Comput\_original2 &= (2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W \\
&+ 2 \times (\frac{2 \times C \times 3 \times 3 \times 3}{C} - 1) \times C \times T \times \frac{H}{2} \times \frac{W}{2} \\
&+ 2 \times (2 \times C \times 1 \times 1 \times 1 - 1) \times \frac{C}{2} \times T \times \frac{H}{2} \times \frac{W}{2} \\
&= C \times T \times \frac{H}{2} \times \frac{W}{2} \times (10C + 101)
\end{aligned} \tag{A3}$$

The computation of improved 3D ShuffleNetV2 (stride = 2):

$$\begin{aligned} Comput\_improved2 &= 2 \times (2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W \\ &+ 2 \times (\frac{2 \times C \times 3 \times 3 \times 3}{C} - 1) \times C \times T \times \frac{H}{2} \times \frac{W}{2} \\ &+ 2 \times (\frac{2 \times C \times 5 \times 5 \times 5}{C} - 1) \times C \times T \times \frac{H}{2} \times \frac{W}{2} \\ &+ 4 \times (2 \times C \times 1 \times 1 \times 1 - 1) \times \frac{C}{4} \times T \times \frac{H}{2} \times \frac{W}{2} \\ &= C \times T \times \frac{H}{2} \times \frac{W}{2} \times (18C + 595) \end{aligned} \tag{A4}$$

It can be seen that the introduction of four branches increases the number of computations, about twice as much.

Appendix B

The computation of the grouping operation and non-grouping operation is calculated as follows.

The computation of the grouping operation: (g groups)

(1) The computation of channel attention:

$$(2 \times \frac{C}{2g} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2g} \times T \times H \times W \tag{A5}$$

(2) The computation of spatial attention:

$$(2 \times \frac{C}{2g} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2g} \times T \times H \times W \tag{A6}$$

(3) The total computation of grouping operation:

$$2 \times (2 \times \frac{C}{2g} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2g} \times T \times H \times W \tag{A7}$$

The computation of the non-grouping operation:

(1) The computation of channel attention:

$$(2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W \tag{A8}$$

(2) The computation of spatial attention:

$$(2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W \tag{A9}$$

(3) The total computation of non-grouping operation:

$$2 \times (2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W \tag{A10}$$

The computation ratio of grouping and non-grouping operation:

$$\frac{group}{non\_group} = \frac{2 \times (2 \times \frac{C}{2g} \times 1 \times 1 \times 1 - 1) \times \frac{C}{2g} \times T \times H \times W}{2 \times (2 \times C \times 1 \times 1 \times 1 - 1) \times C \times T \times H \times W} \approx \frac{1}{4g^2} \tag{A11}$$

It can see that the grouping operation can reduce the number of computations. The computation of grouping operation is about $\frac{1}{4g^2}$ of that of the non-grouping operation.

References

1. Zhang, J.; Zheng, Z.; Xie, X.; Gui, Y.; Kim, G.J. ReYOLO: A traffic sign detector based on network reparameterization and features adaptive weighting. *J. Ambient. Intell. Smart Environ.* **2022**, *14*, 317–334. [CrossRef]
2. Zhang, J.; Zou, X.; Kuang, L.D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A more comprehensive traffic sign detection benchmark. *Hum.-Centric Comput. Inf. Sci.* **2022**, *12*, 23.
3. Zhang, Z.; Zhang, L. Unsupervised Pixel-Level Detection of Rail Surface Defects Using Multistep Domain Adaptation. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 5784–5795. [CrossRef]
4. Wei, D.; Wei, X.; Tang, Q.; Jia, L.; Yin, X.; Ji, Y. RTLSeg: A novel multi-component inspection network for railway track line based on instance segmentation. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105822. [CrossRef]
5. Liu, Y.; Xiao, H.; Xu, J.; Zhao, J. A rail surface defect detection method based on pyramid feature and lightweight convolutional neural network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10. [CrossRef]
6. Yang, H.; Wang, Y.; Hu, J.; He, J.; Yao, Z.; Bi, Q. Segmentation of track surface defects based on machine vision and neural networks. *IEEE Sens. J.* **2021**, *22*, 1571–1582. [CrossRef]
7. Su, S.; Du, S.; Lu, X. Geometric Constraint and Image Inpainting-Based Railway Track Fastener Sample Generation for Improving Defect Inspection. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23883–23895. [CrossRef]
8. Bai, T.; Yang, J.; Xu, G.; Yao, D. An optimized railway fastener detection method based on modified Faster R-CNN. *Measurement* **2021**, *182*, 109742. [CrossRef]
9. Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; pp. 65–72.
10. 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]
11. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 3551–3558.
12. Peng, X.; Zou, C.; Qiao, Y.; Peng, Q. Action recognition with stacked fisher vectors. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 581–595.
13. Nazir, S.; Yousaf, M.H.; Velastin, S.A. Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition. *Comput. Electr. Eng.* **2018**, *72*, 660–669. [CrossRef]
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
17. Vahdani, E.; Tian, Y. Deep learning-based action detection in untrimmed videos: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4302–4320. [CrossRef] [PubMed]
18. Gu, C.; Sun, C.; Ross, D.A.; Vondrick, C.; Pantofaru, C.; Li, Y.; Vijayanarasimhan, S.; Toderici, G.; Ricco, S.; Sukthankar, R.; et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6047–6056.
19. 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.
20. Sun, C.; Shrivastava, A.; Vondrick, C.; Murphy, K.; Sukthankar, R.; Schmid, C. Actor-centric relation network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 318–334.
21. Yang, X.; Yang, X.; Liu, M.Y.; Xiao, F.; Davis, L.S.; Kautz, J. Step: Spatio-temporal progressive learning for video action detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 264–272.
22. Wu, C.Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; Girshick, R. Long-term feature banks for detailed video understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 284–293.
23. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
24. Wu, J.; Kuang, Z.; Wang, L.; Zhang, W.; Wu, G. Context-aware rcnn: A baseline for action detection in videos. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 440–456.

25. Pan, J.; Chen, S.; Shou, M.Z.; Liu, Y.; Shao, J.; Li, H. Actor-context-actor relation network for spatio-temporal action localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 464–474.
26. Chen, S.; Sun, P.; Xie, E.; Ge, C.; Wu, J.; Ma, L.; Shen, J.; Luo, P. Watch only once: An end-to-end video action detection framework. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8178–8187.
27. Sui, L.; Zhang, C.L.; Gu, L.; Han, F. A Simple and Efficient Pipeline to Build an End-to-End Spatial-Temporal Action Detector. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 5999–6008.
28. Chang, S.; Wang, P.; Wang, F.; Feng, J.; Shou, M.Z. DOAD: Decoupled One Stage Action Detection Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 3122–3131.
29. 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.
30. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
31. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
32. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
33. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
34. Hu, X.; Wang, T.; Huang, J.; Peng, T.; Liu, J.; He, R. Subway Driver Behavior Detection Method Based on Multi-features Fusion. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 3378–3385.
35. 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.
36. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
37. Suo, D.; Wei, X.; Wei, D. Gesture Recognition of Subway Drivers Based on Improved Dense Trajectory Algorithm. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1554–1559.
38. Zhang, Q.L.; Yang, Y.B. Sa-net: Shuffle attention for deep convolutional neural networks. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2235–2239.
39. 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.
40. Yang, F. A Multi-Person Video Dataset Annotation Method of Spatio-Temporally Actions. *arXiv* **2022**, arXiv:2204.10160.
41. Jocher, G.; Changyu, L.; Hogan, A.; Yu, L.; Rai, P.; Sullivan, T. Ultralytics/yolov5: Initial Release. 2020. Available online: https://zenodo.org/records/3908560 (accessed on 1 February 2023).
42. Dutta, A.; Gupta, A.; Zissermann, A. VGG Image Annotator (VIA). 2016. Available online: https://www.robots.ox.ac.uk/~vgg/software/via/ (accessed on 2 February 2023).

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

Article

Hubble Meets Webb: Image-to-Image Translation in Astronomy

Vitaliy Kinakh [1], Yury Belousov [1], Guillaume Quétant [1], Mariia Drozdova [1], Taras Holotyak [1], Daniel Schaerer [2] and Slava Voloshynovskiy [1,*]

[1] Department of Computer Science, University of Geneva, 1227 Carouge, Switzerland; vitaliy.kinakh@unige.ch (V.K.); yury.belousov@unige.ch (Y.B.); guillaume.quetant@unige.ch (G.Q.); mariia.drozdova@unige.ch (M.D.); taras.holotyak@unige.ch (T.H.)

[2] Department of Astronomy, University of Geneva, 1290 Versoix, Switzerland; daniel.schaerer@unige.ch

* Correspondence: svolos@unige.ch

Abstract: This work explores the generation of James Webb Space Telescope (JWSP) imagery via image-to-image translation from the available Hubble Space Telescope (HST) data. Comparative analysis encompasses the Pix2Pix, CycleGAN, TURBO, and DDPM-based Palette methodologies, assessing the criticality of image registration in astronomy. While the focus of this study is not on the scientific evaluation of model fairness, we note that the techniques employed may bear some limitations and the translated images could include elements that are not present in actual astronomical phenomena. To mitigate this, uncertainty estimation is integrated into our methodology, enhancing the translation's integrity and assisting astronomers in distinguishing between reliable predictions and those of questionable certainty. The evaluation was performed using metrics including MSE, SSIM, PSNR, LPIPS, and FID. The paper introduces a novel approach to quantifying uncertainty within image translation, leveraging the stochastic nature of DDPMs. This innovation not only bolsters our confidence in the translated images but also provides a valuable tool for future astronomical experiment planning. By offering predictive insights when JWST data are unavailable, our approach allows for informed preparatory strategies for making observations with the upcoming JWST, potentially optimizing its precious observational resources. To the best of our knowledge, this work is the first attempt to apply image-to-image translation for astronomical sensor-to-sensor translation.

Keywords: image-to-image translation; denoising diffusion probabilistic models; uncertainty estimation; satellite image generation; image registration

Citation: Kinakh, V.; Belousov, Y.; Quétant, G.; Drozdova, M.; Holotyak, T.; Schaerer, D.; Voloshynovskiy, S. Hubble Meets Webb: Image-to-Image Translation in Astronomy. *Sensors* **2024**, *24*, 1151. https://doi.org/10.3390/s24041151

Academic Editors: Dae-Ki Kang and Sukho Lee

Received: 1 January 2024
Revised: 4 February 2024
Accepted: 7 February 2024
Published: 9 February 2024

1. Introduction

In this paper, we explore the problem of predicting the visible sky images captured by the James Webb Space Telescope (JWST), hereafter referred to as 'Webb' [1], using the available data from the Hubble Space Telescope (HST), hereinafter called 'Hubble' [2]. There is much interest in this type of problem in fields such as astrophysics, astronomy, and cosmology, encompassing a variety of data types and sources. This includes the translation of observations of galaxies in visible light [3] and predictions of dark matter [4]. The data registered from different sources may be acquired at different times, by different sensors, in different bands, with different resolutions, sensitivities, and levels of noise. The exact underlying mathematical model for transforming data between these sources is very complex and largely unknown. Thus, we will try to address this problem based on an image-to-image translation approach.

Despite the great success of image-to-image translation in computer vision, its adoption in the astrophysics community has been limited, even though there is a lot of data available for such tasks that might enable sensor-to-sensor translation, conversion between different spectral bands, and adaptation among various satellite systems.

Before the launch of missions such as Euclid [5], the radio telescope Square Kilometre Array [6], and others, there has been a significant interest in advancing image-to-image translation techniques for astronomical data to: (i) enable efficient mission planning due to the high complexity and cost of exhaustive space exploration, allowing for the prioritization of specific space regions using existing data; and (ii) generate sufficient synthetic data for machine learning (ML) analysis as soon as the first real images from new imaging missions are available in adequate quantities.

We focus on the images collected by both the Hubble and the Webb telescopes, taken at different times, as illustrated in Figure 1. Thus, we present our work as a proof-of-concept for image-to-image translation, aiming to predict Webb telescope images using those from Hubble. This technique, once validated, could inform the planning of future missions and experiments by enabling the prediction of Webb telescope observations from existing Hubble data.

We assume that, despite the time lapse between Hubble's and Webb's data acquisition, the astronomical scenes of interest have remained relatively stable, conforming to the slow-changing physics of the observed phenomena. However, there is a substantial disparity in the imaging technologies of the two telescopes, affecting not only resolution and signal-to-noise ratio but also the visual representation of the phenomena due to different underlying physical principles and the images being taken at various wavelengths.

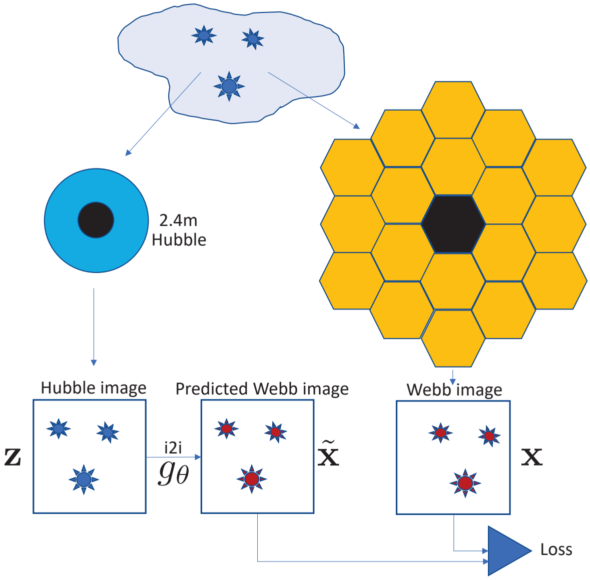


Figure 1. Image-to-image astronomical setup under study. Given two imaging systems, Hubble and Webb, characterized by different bands, resolutions, orbits, and time of image acquisition, the problem is to predict the Webb images $\tilde{\mathbf{x}}$ as close as possible to the original Webb images $\mathbf{x}$ from the Hubble ones $\mathbf{z}$ using a learnable model g_θ. The considered setup is paired but is characterized by inaccurate geometrical synchronization between the paired images.

Our study reveals that Hubble and Webb data are typically dis-synchronized by approximately 3–5 pixels, a discrepancy mainly attributed to synchronization with respect to celestial coordinates during Webb's data pre-processing and differing resolutions. Although this misalignment is subtle to the naked eye, we found that it significantly impairs the accuracy of paired image-to-image translation, highlighting the critical need for precise data alignment. To address this problem, we introduce two synchronization methods using computer vision keypoints and descriptors: (a) global synchronization applies a single affine transformation to the entire image; (b) local synchronization divides the image into patches and computes individual affine transformations for each patch. We compare the impact on the performance of image-to-image translation when using these synchronization methods against provided synchronization with respect to celestial coordinates.

We compare several types of image-to-image translation methods: (i) fully paired methods such as Pix2Pix [7] and their variations; (ii) fully unpaired methods such as CycleGAN [8]; (iii) hybrid methods that can be used for both fully paired setups, fully unpaired setups, or setups where part of the data is paired, and part of the data is unpaired, as advocated by the TURBO approach [9]; (iv) denoising diffusion probabilistic models (DDPM) [10] based image-to-image translation method Palette [11]. We investigate the influence of pairing and different types of synchronization for the above methods. We demonstrate that paired methods produce results superior to unpaired ones. At the same time, the paired methods Pix2Pix and TURBO are subject to the accuracy of synchronization. Local synchronization produces the most accurate translation results, according to several metrics of performance.

Furthermore, we show that there is a high potential for uncertainty in the estimation when using DDPM models for image-to-image translation since they can produce multiple outputs for one input. This stochastic translation enabled us to establish the regions that appear to be very stable in each run and the ones that are characterized by high variability.

In summary, we run experiments for image-to-image translation on non-synchronized, globally synchronized, and locally synchronized Hubble–Webb pairs. We report the results using multiple metrics: MSE, SSIM [12], PSNR, LPIPS [13], and FID [14]. We use computer vision-based metrics since we are working with telescope images represented as RGB images.

The main focus of this paper is not on the scientific inquiry into the fairness of predictive models. We acknowledge that our results, generated through the image-to-image translation technique, are subject to limitations inherent to such approaches. The data and methods utilized may not be exhaustive or infallible, and the results should therefore be interpreted with caution, as they are not immune to inaccuracies and may contain hallucinated elements which do not correspond to real astronomical phenomena.

Therefore, to enhance the integrity of the image-to-image translation provided in this study, we incorporate uncertainty estimation into our methodology. This feature is designed to assist astronomers by delineating areas within the translated images where the model's predictions are reliable from those where the certainty of prediction remains questionable. Such delineation is crucial in guiding astronomers to discern between regions of high confidence and those that require further scrutiny or could potentially mislead them.

The proposed approach, with its ability to estimate uncertainty, may serve as an instrumental tool for planning future astronomical experiments. In scenarios where observational data from the Webb telescope are not yet available, our model can offer predictive insights based on existing Hubble Space Telescope data. This capability acts as a provisional glimpse into the future, enabling researchers to strategize upcoming observations with the Webb telescope, potentially optimizing the allocation of its valuable observational time.

Our contributions include: (i) the introduction of image-based synchronization for astrophysics data in view of image-to-image translation problems; (ii) a comparison of the image-to-image translation methods for Hubble to Webb translation, and a study of the effect of synchronization on different models; (iii) the introduction of an innovative way of uncertainty estimation in probabilistic inverse solvers or translation methods based on denoising diffusion probabilistic models. In summary, **our main contribution** is: the demonstration of the potential of using deep learning-based image-to-image translation in astronomical imaging, exemplified by Hubble to Webb image translation.

2. Related Work

2.1. Comparison between Webb and Hubble Telescopes

In Figures 2 and 3, the same part of the sky captured by the Hubble and Webb telescopes is shown in the RGB format. The main differences between the Hubble and Webb telescopes are: (i) **Spatial resolution**—The Webb telescope, featuring a 6.5-m primary mirror, offers superior resolution compared to Hubble's 2.4-m mirror, which is particularly noticeable in infrared observations [15]. This enables Webb to capture images of objects up to 100 times fainter than Hubble, as evident in the central spiral galaxy in Figure 3.

(ii) **Wavelength coverage**— Hubble, optimized for ultraviolet and visible light (0.1 to 2.5 microns), contrasts with Webb's focus on infrared wavelengths (0.6 to 28.5 microns) [16]. While this differentiation allows Webb to observe more distant and fainter celestial objects, including the earliest stars and galaxies, it is crucial to note that the IR emission captured by Webb differs inherently from the UV or visible light observed by Hubble. The distinction is not solely in the resolution or sensitivity between the Hubble Space Telescope (HST) and the James Webb Space Telescope (JWST) but also in the varying absorption of light by dust within different galaxy types. However, our proposed image-to-image translation method does not aim to delve into these observational differences. Instead, our focus is to explore whether image-to-image translation can effectively simulate Webb telescope imagery based on the existing data from Hubble. This approach seeks to leverage the available Hubble data to anticipate and interpret the observations that Webb might deliver, without directly analyzing the spectral and compositional differences between the images captured by the two telescopes.

(iii) **Light-collecting capacity**—Webb's substantially larger mirror provides over six times the light-collecting area compared to Hubble, essential for studying longer, dimmer wavelengths of light from distant, redshifted objects [15]. This is exemplified in Webb's images, which reveal smaller galaxies and structures not visible in Hubble's observations, highlighted in yellow in Figure 3.

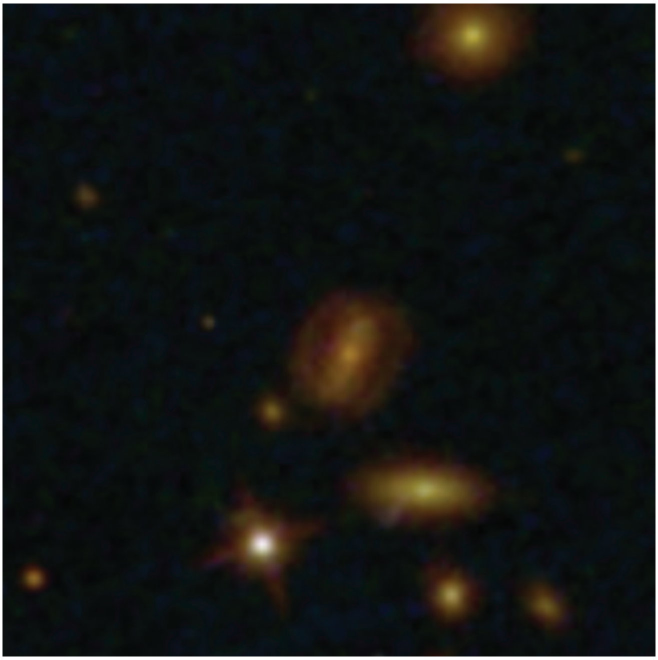


Figure 2. Hubble photo of Galaxy Cluster SMACS 0723 [17].

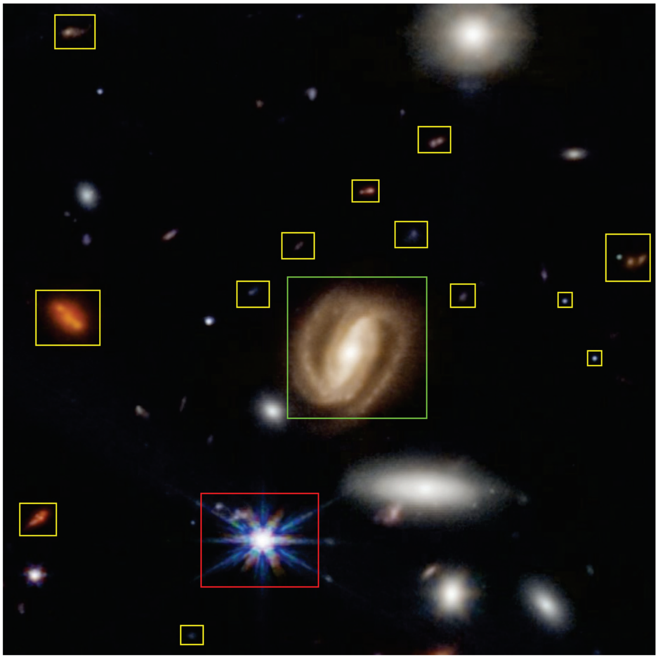


Figure 3. Webb image of Galaxy Cluster SMACS 0723 [18].

2.2. Image-to-Image Translation

Image-to-image translation [19] is the task of transforming an image from one domain to another, where the goal is to understand the mapping between an input image and an output image. Image-to-image translation methods have shown great success in computer vision tasks, including transferring different styles [20], colorization [21], superresolution [22], visible to infrared translation [23], and many others [24]. There are two types of image-to-image translation methods: *unpaired* [25] (sometimes called unsupervised) and *paired* [26]. Unpaired setups do not require fixed pairs of corresponding images, while paired setups do. In this paper, we also introduce a hybrid method for image-to-image translation, called TURBO [9], which is a generalization of the above-mentioned paired and unpaired setups and provides an information–theoretic interpretation of this method. For the completeness of our study, we also consider newly introduced denoising diffusion probabilistic models (DDPM) as image-to-image translation models [11].

2.3. Image-to-Image Translation in Astrophysics

Image-to-image translation has been used in astrophysics for galaxy simulation [3], but these methods have mostly been used for denoising [27] optical and radio astrophysical data [28]. The task of predicting the images of one telescope from another using image-to-image translation remains largely under-researched.

2.4. Metrics

The following metrics were used to evaluate the quality of the generated images:

- Mean square error (MSE) between the original and the generated Webb images;
- To address an issue that the MSE is not highly indicative of the perceived similarity of images, we calculate the Structural Similarity Index (SSIM) [12] between the original and generated Webb images;
- Fréchet Inception Distance (FID): proposed in [14]. Instead of a simple pixel-by-pixel comparison of images, FID estimates the mean and standard deviation of one of the

deep layers in the pretrained convolutional neural network. It has become one of the most widely used metrics for the image-to-image translation task;

- Peak Signal-to-Noise Ratio (PSNR): This metric evaluates the quality of the generated images by comparing the maximum possible power of a signal (original images) to the power of the same images after distortion (generated images). PSNR is often used as a measure of reconstruction quality in image compression and restoration tasks;
- Learned Perceptual Image Patch Similarity (LPIPS): proposed in [13]. LPIPS measures the perceptual similarity between images by using deep features extracted from a pretrained neural network. It is designed to better reflect human perception of image similarity compared to traditional metrics like MSE or PSNR.

3. Proposed Approach

3.1. Dataset

We use images from the Hubble and Webb telescopes as the dataset. In particular, we use images of Galaxy Cluster SMACS 0723 [29]. An example of the image is shown in Figure 2. For the Webb, we use post-processed NIRCam images [30], available as RGB images, provided by ESA/NASA/STScI. Webb images are available publicly at [17]. We then select the corresponding Hubble images [18]. Since the Hubble images are smaller than Webb images, we upsampled them using bicubic interpolation for comparison purposes.

3.2. Image Registration

Image registration or synchronization is needed to ensure that pixels in different data sources represent the same position in observed space. Even though astronomical data are generally synchronized, there is always room for synchronization improvement, especially at the local level. In this section, we compare three synchronization setups for Hubble to Webb translation: synchronization with respect to celestial coordinates, algorithmic or automated global synchronization, and local synchronization, as schematically shown in Figure 4.

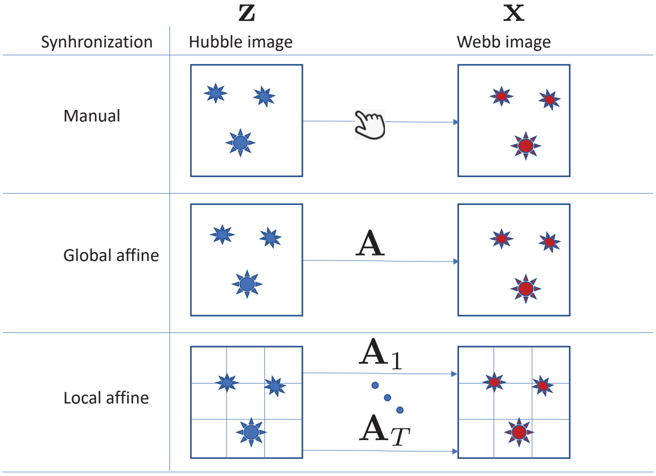


Figure 4. Synchronization setups under investigation in paired image-to-image translation problems: synchronization with respect to celestial coordinates; global synchronization, when images are matched via a global affine transform $\mathbf{A}$; and local synchronization, when images are divided into local blocks and matched via a set of local affine transforms $\mathbf{A}_i$, $1 \leq i \leq T$.

Synchronization with respect to celestial coordinates. In this setup, the data are used directly with the provided synchronization with respect to celestial coordinates.

Global synchronization. The data are synchronized using SIFT [31] feature descriptors and the RANSAC [32] matching algorithm. The feature descriptors are computed for the entire image from both the Hubble and Webb telescopes.

Local synchronization. The data are synchronized using SIFT feature descriptors and the RANSAC matching algorithm, with the feature descriptors being computed from image patches. Specifically, input images from both the Hubble and Webb telescopes are divided into a grid made of nine patches, arranged in a three × three configuration both vertically and horizontally, before the cropping process.

The non-synchronized and synchronized Webb and Hubble images can be viewed in our demo: hubble-to-webb.herokuapp.com (accessed on 8 February 2024).

3.3. TURBO

3.3.1. Mathematical Interpretation

The TURBO framework [9] is based on an auto-encoder (AE) structure and is represented by an encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ that are deep networks parametrized by the parameters ϕ and θ, respectively. A block diagram for the TURBO system is shown in Figure 5.

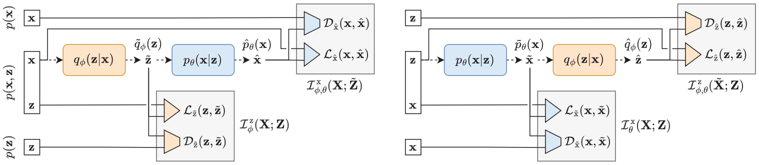

$p(\mathbf{x})$
$\mathbf{x}$
$p(\mathbf{x},\mathbf{z})$
$\mathbf{x}$
$\mathbf{z}$
$p(\mathbf{z})$
$\mathbf{z}$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\tilde{q}_\phi(\mathbf{z})$
$\tilde{\mathbf{z}}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$\hat{p}_\theta(\mathbf{x})$
$\hat{\mathbf{x}}$
$\mathcal{D}_{\hat{\mathrm{x}}}(\mathbf{x},\hat{\mathbf{x}})$
$\mathcal{L}_{\hat{\mathrm{x}}}(\mathbf{x},\hat{\mathbf{x}})$
$\mathcal{I}^{\mathrm{x}}_{\phi,\theta}(\mathbf{X};\tilde{\mathbf{Z}})$
$\mathcal{L}_{\tilde{\mathrm{z}}}(\mathbf{z},\tilde{\mathbf{z}})$
$\mathcal{D}_{\tilde{\mathrm{z}}}(\mathbf{z},\tilde{\mathbf{z}})$
$\mathcal{I}^{\mathrm{z}}_{\phi}(\mathbf{X};\mathbf{Z})$
$\mathbf{z}$
$\mathbf{z}$
$\mathbf{x}$
$\mathbf{x}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$\tilde{p}_\theta(\mathbf{x})$
$\tilde{\mathbf{x}}$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\hat{q}_\phi(\mathbf{z})$
$\hat{\mathbf{z}}$
$\mathcal{D}_{\hat{\mathrm{z}}}(\mathbf{z},\hat{\mathbf{z}})$
$\mathcal{L}_{\hat{\mathrm{z}}}(\mathbf{z},\hat{\mathbf{z}})$
$\mathcal{I}^{\mathrm{z}}_{\phi,\theta}(\tilde{\mathbf{X}};\mathbf{Z})$
$\mathcal{L}_{\tilde{\mathrm{x}}}(\mathbf{x},\tilde{\mathbf{x}})$
$\mathcal{D}_{\tilde{\mathrm{x}}}(\mathbf{x},\tilde{\mathbf{x}})$
$\mathcal{I}^{\mathrm{x}}_{\theta}(\mathbf{X};\mathbf{Z})$

Figure 5. TURBO scheme: direct (**left**) and reverse (**right**) paths.

According to the framework we used, given a pair of data samples (Hubble and Webb images) $(\mathbf{x},\mathbf{z}) \sim p(\mathbf{x},\mathbf{z})$, where $\mathbf{z}$ is a Hubble image and $\mathbf{x}$ is a Webb image, the system maximizes the mutual information between $\mathbf{x}$ and $\mathbf{z}$ for both encoder and decoder in *direct* and *reverse* paths.

Two approximations of the joint distribution can be defined as follow:

$$q_\phi(\mathbf{x},\mathbf{z}) := q_\phi(\mathbf{z}|\mathbf{x})\ \overbrace{p(\mathbf{x})}^{\substack{\text{real}\\\text{data}}} = q_\phi(\mathbf{x}|\mathbf{z})\ \overbrace{\tilde{q}_\phi(\mathbf{z})}^{\substack{\text{synthetic}\\\text{data}}}\ , \tag{1}$$

$$p_\theta(\mathbf{x},\mathbf{z}) := \underbrace{p_\theta(\mathbf{x}|\mathbf{z})}_{\substack{\text{known}\\\text{networks}}}\ p(\mathbf{z}) = \underbrace{p_\theta(\mathbf{z}|\mathbf{x})}_{\substack{\text{unknown}\\\text{networks}}}\ \tilde{p}_\theta(\mathbf{x})\ , \tag{2}$$

the marginal distributions are approximated through reparametrizations involving unknown networks. These are represented as $\tilde{q}_\phi(\mathbf{z}) = \int q_\phi(\mathbf{x},\mathbf{z})\mathrm{d}\mathbf{x}$ and $\tilde{p}_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x},\mathbf{z})\mathrm{d}\mathbf{z}$, relating to the synthetic variables in latent spaces. Furthermore, in our work, we also utilize two approximated marginal distributions for the reconstructed synthetic variables in spaces, denoted as $\hat{q}_\phi(\mathbf{z}) = \int \tilde{p}_\theta(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})\mathrm{d}\mathbf{x}$ and $\hat{p}_\theta(\mathbf{x}) = \int \tilde{q}_\phi(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})\mathrm{d}\mathbf{z}$.

The variational approximation is considered for the *direct path* of the TURBO system based on the maximization of two bounds on mutual information for the latent space and the reconstruction space:

$$\mathcal{I}(\mathbf{X};\mathbf{Z}) = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}\right] \geq \underbrace{\mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right]}_{-\mathcal{L}_{\tilde{\mathrm{z}}}(\mathbf{z},\tilde{\mathbf{z}})} - \underbrace{D_{\mathrm{KL}}\left(p(\mathbf{z})\|\tilde{q}_\phi(\mathbf{z})\right)}_{\mathcal{D}_{\tilde{\mathrm{z}}}(\mathbf{z},\tilde{\mathbf{z}})}, \tag{3}$$

$$\mathcal{I}_{\phi}(\mathbf{X};\tilde{\mathbf{Z}}) = \mathbb{E}_{q_{\phi}(\mathbf{x},\mathbf{z})}\left[\log \frac{q_{\phi}(\mathbf{x},\mathbf{z})}{p(\mathbf{x})\tilde{q}_{\phi}(\mathbf{z})}\right] \geq \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{x},\mathbf{z})}[\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{-\mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{x})\|\hat{p}_{\theta}(\mathbf{x}))}_{\mathcal{D}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})}. \tag{4}$$

Thus, the network is trained in such a way to maximize a weighted sum of (3) and (4) in order to find the best parameters ϕ and θ of the encoder and the decoder, respectively. This is achieved in the *direct path* by minimising the $\mathcal{L}^{\text{direct}}$ loss, representing the left network shown in Figure 5:

$$\mathcal{L}^{\text{direct}}(\phi,\theta) = \mathcal{L}_{\tilde{\mathbf{z}}}(\mathbf{z},\tilde{\mathbf{z}}) + \mathcal{D}_{\tilde{\mathbf{z}}}(\mathbf{z},\tilde{\mathbf{z}}) + \lambda_D\mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}}) + \lambda_D\mathcal{D}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}}), \tag{5}$$

where $\mathbf{z}$ is real Hubble image, $\mathbf{x}$ is real Webb image, $\tilde{\mathbf{z}}$ predicted Hubble image generated by $q_{\phi}(\mathbf{z}|\mathbf{x})$ from real Webb image $\mathbf{x}$, $\hat{\mathbf{x}}$ is Webb image reconstructed from generated Hubble image $\tilde{\mathbf{z}}$, $\mathcal{L}_{\tilde{\mathbf{z}}}(\mathbf{z},\tilde{\mathbf{z}})$ reconstruction loss between real and generated Hubble images, $\mathcal{D}_{\tilde{\mathbf{z}}}(\mathbf{z},\tilde{\mathbf{z}})$ discriminator loss for generated Hubble images, $\mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})$ present cycle reconstruction loss between real and reconstructed Webb images, $\mathcal{D}_{\hat{\mathbf{x}}}(\mathbf{x},\hat{\mathbf{x}})$ is discriminator loss in the reconstructed Webb images, and λ_D is a parameter controlling the trade-off between the terms in (3) and (4).

The variational approximation for the *reverse path* is:

$$\mathcal{I}(\mathbf{X};\mathbf{Z}) = \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}\right] \geq \underbrace{\mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{-\mathcal{L}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{x})\|\tilde{p}_{\theta}(\mathbf{x}))}_{\mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}})}, \tag{6}$$

$$\mathcal{I}_{\theta}(\tilde{\mathbf{X}};\mathbf{Z}) = \mathbb{E}_{p_{\theta}(\mathbf{x},\mathbf{z})}\left[\log \frac{p_{\theta}(\mathbf{x},\mathbf{z})}{\tilde{p}_{\theta}(\mathbf{x})p(\mathbf{z})}\right] \geq \underbrace{\mathbb{E}_{p_{\theta}(\mathbf{x},\mathbf{z})}[\log q_{\phi}(\mathbf{z}|\mathbf{x})]}_{-\mathcal{L}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}})} - \underbrace{D_{\mathrm{KL}}(p(\mathbf{z})\|\hat{q}_{\phi}(\mathbf{z}))}_{\mathcal{D}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}})}. \tag{7}$$

The *reverse path* loss $\mathcal{L}^{\text{reverse}}(\phi,\theta)$ is represented by the right network shown in Figure 5:

$$\mathcal{L}^{\text{reverse}}(\phi,\theta) = \mathcal{L}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}}) + \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}}) + \lambda_R\mathcal{L}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}}) + \lambda_R\mathcal{D}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}}), \tag{8}$$

where $\tilde{\mathbf{x}}$ is a Webb image, generated by $p_{\theta}(\mathbf{x}|\mathbf{z})$ from a real Hubble image $\mathbf{z}$, $\hat{\mathbf{z}}$ is a Hubble image reconstructed from generated Webb image $\tilde{\mathbf{x}}$, $\mathcal{L}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}})$ is reconstruction loss between the real and generated Webb images, $\mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}})$ is discriminator loss in the generated Webb images, $\mathcal{L}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}})$ is cycle reconstruction loss between real and reconstructed Hubble images, $\mathcal{D}_{\hat{\mathbf{z}}}(\mathbf{z},\hat{\mathbf{z}})$ discriminator loss in the reconstructed Hubble images, and λ_R is a parameter controlling the trade-off between (6) and (7).

A detailed derivation and analysis of TURBO can be found in [9].

The TURBO method is versatile and adaptable to various setups. It supports a fully paired configuration, utilizing direct and reverse path losses, provided above, which are applicable when data pairs are fully accessible during training. In cases where such pairs are unavailable for training, an unpaired configuration is viable. Additionally, a mixed setup can be employed, combining both paired and unpaired data. This method imposes no constraints on the architecture of the encoder and decoder, offering a broad range of architectural choices.

3.3.2. Paired Setup: Pix2Pix as Particular Case of TURBO

Pix2Pix [7] image-to-image translation method can be viewed as a paired case of TURBO approach, with only reverse path, where $\lambda_R = 0$ in (9):

$$\mathcal{L}^{Pix2Pix}(\theta) = \mathcal{L}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}}) + \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x},\tilde{\mathbf{x}}). \tag{9}$$

Thus, the direct path is not used as the training of the encoder–decoder pair and Pix2Pix uses uses the deterministic decoder $\tilde{\mathbf{x}} = g_\theta(\mathbf{z})$.

3.3.3. Unpaired Setup: CycleGAN as Particular Case of TURBO

The CycleGAN [8] image-to-image translation method can be viewed as a particular case of the TURBO approach, with both a direct and reverse path, with cycle reconstruction losses and discriminator losses for predicted images, with:

$$\mathcal{L}^{\text{CycleGAN}}(\phi, \theta) = \mathcal{D}_{\tilde{\mathbf{z}}}(\mathbf{z}, \tilde{\mathbf{z}}) + \lambda_D \mathcal{L}_{\hat{\mathbf{x}}}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_T \mathcal{D}_{\tilde{\mathbf{x}}}(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda_T \lambda_R \mathcal{L}_{\hat{\mathbf{z}}}(\mathbf{z}, \hat{\mathbf{z}}), \tag{10}$$

CycleGAN does not have paired components in the latent space in comparison to TURBO.

3.4. Denoising Diffusion Based Image-to-Image Translation

Conditional denoising diffusion probabilistic models [10] for image-to-image translation apply a denoising process that is conditioned on the input image [11]. Image-to-image diffusion models are conditional models of the form $p_\theta(\mathbf{x}|\mathbf{z})$, where $\mathbf{x}$ is a generated Webb image, and $\mathbf{z}$ is a Hubble image, used as a condition. In fact, the DDPM models are derived from the Variational Autoencoder [33] with the decomposition of the latent space of $\mathbf{z}$ as a hierarchical Markov model $\mathbf{z}_T \to \mathbf{z}_{T-1} \to \cdots \to \mathbf{z}_0$ [34].

In practice, the conditional image is concatenated to the input noisy image. During training, detailed in Algorithm 1, we use a simple DDPM training loss (11):

$$\mathcal{L}^{DDPM}(\theta) = \mathbb{E}_{t,\mathbf{z},\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \mathbf{z}, t\right)\right\|^2\right], \tag{11}$$

where $\mathbf{x}_0$ is Webb image, $\mathbf{z}$ is the input Hubble image, used in conditioning, $\boldsymbol{\epsilon}$ is Gaussian zero mean unit variance noise added at step t, $\boldsymbol{\epsilon}_\theta$ is conditional DDPM, and $\bar{\alpha}_t$ is noise scale parameter, added at step t.

Algorithm 1 Training a denoising model $\boldsymbol{\epsilon}_\theta$

```
Define noise schedule $\beta_1, \beta_2, \ldots, \beta_T$
Compute $\bar{\alpha}_t$ for $t = 1$ to $T$ using $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$
repeat
    $(\mathbf{x}, \mathbf{z}) \sim p(\mathbf{x}, \mathbf{z})$
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $t \sim 1 \ldots T$
    Take a gradient descent step on $\nabla_\theta \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \mathbf{z}, t\right)\right\|^2$
until converged
```

In the inference phase of the conditional denoising diffusion probabilistic model, detailed in Algorithm 2, the model starts with an initial noisy sample $\mathbf{x}_T$ from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$; then, the model utilizes a learned denoising function $\boldsymbol{\epsilon}_\theta$, which incorporates the conditioning Hubble image $\mathbf{x}$, to iteratively denoise the image at each timestep t. The image is updated according to (12):

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \frac{1-\bar{\alpha}_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)\right) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{12}$$

where $\boldsymbol{\epsilon}$ is sampled from Gaussian noise. This denoising process is repeated for T steps until the final image $\mathbf{x}_0$ is obtained.

Algorithm 2 Inference in T iterative refinement steps

```
$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, \ldots, 1$ do
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\boldsymbol{\epsilon} = 0$
    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \frac{1-\bar{\alpha}_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)\right) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
end for
return $\mathbf{x}_0$
```

4. Uncertainty Estimation

In this section, we show how denoising diffusion probabilistic models can be used for the prediction of uncertainty maps. By design, DDPMs are stochastic generators at each sampling step, so it is possible to generate multiple predictions for the same input. The ensemble of predictions allows us to compute the pixel-wise deviation maps that visualize the uncertainty of the predictions. In Figure 6, we display the *true uncertainty map* $\mathbf{U}$, computed as $\sqrt{\frac{\sum_{i=1}^{N}(\hat{\mathbf{x}}_i - \mathbf{x})^2}{N}}$, where $\mathbf{x}$ is the target Webb image, $\hat{\mathbf{x}}_i$ is the i-th predicted Webb image, $\bar{\hat{\mathbf{x}}}$ is the averaged predicted image estimated from $\hat{\mathbf{x}}_i$, and N is the number of generated images. In our experiments, we have used 100 generations to compute the *estimated uncertainty map* $\hat{\mathbf{U}}$, computed as $\sqrt{\frac{\sum_{i=1}^{N}\left(\hat{\mathbf{x}}_i - \bar{\hat{\mathbf{x}}}\right)^2}{N}}$.

The uncertainty map can be used for analyzing and evaluating the DDPM results by indicating the regions of low and high variability as a measure of uncertainty in each experiment. It is remarkable that this approach is very discriminating for the different types of space objects: point objects (shown in Figures 7–9), galaxies (shown in Figure 8), and stars (shown in Figure 9). Furthermore, we have found that the method is able to detect the presence of point source objects in the estimated uncertainty maps, while such objects were not usually directly detectable in the Hubble images or in the predicted Webb images (highlighted with orange boxes in Figures 7 and 9). The point sources that were not present in the Hubble images were not completely predicted in the Webb images when considering these images independently. However, the use of an uncertainty map allowed us to spot their presence in the uncertainty maps, which are highlighted with red boxes in the above-mentioned figures. To further evaluate the performance, we introduce the Peak Signal-to-Uncertainty Ratio (PSUR), computed as $\mathrm{PSUR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_{\mathbf{x}}^2}{\mathrm{mean}(\hat{\mathbf{U}})}\right)$ dB, where $\mathrm{MAX}_{\mathbf{x}}$ is the maximum possible pixel value of the image. This metric, analogous to PSNR but using the uncertainty map instead of MSE, offers a measure of how distinguishable the true signal is from the uncertainty inherent in the prediction process. We compute PSUR value for every uncertainty map, shown in Figures 6–9.

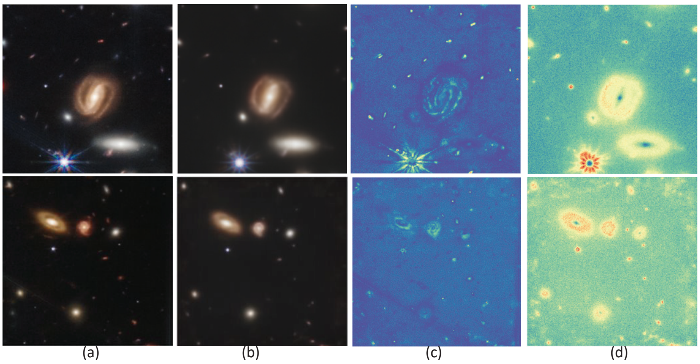


Figure 6. Uncertainty map visualization. (**a**) $\mathbf{x}$ target Webb image, (**b**) $\bar{\hat{\mathbf{x}}}$ predicted image, averaged from $\hat{\mathbf{x}}_i$, (**c**) true uncertainty, (**d**) estimated uncertainty. The estimated PSUR: 28.99 dB.

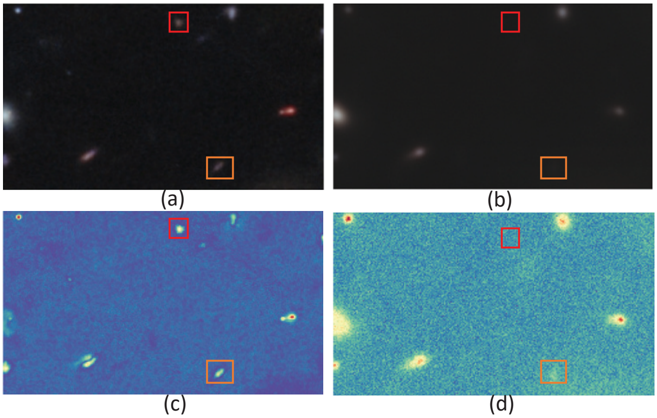


Figure 7. Uncertainty map for point sources: (**a**) target Webb image; (**b**) predicted Webb image; (**c**) true uncertainty; (**d**) estimated uncertainty. The point sources, that were missed, and for which there is no sign in the uncertainty map, are highlighted with a red box. The point sources are missed, but for which there is a sign in the uncertainty map, are highlighted with an orange box. The estimated PSUR: 26.72 dB.

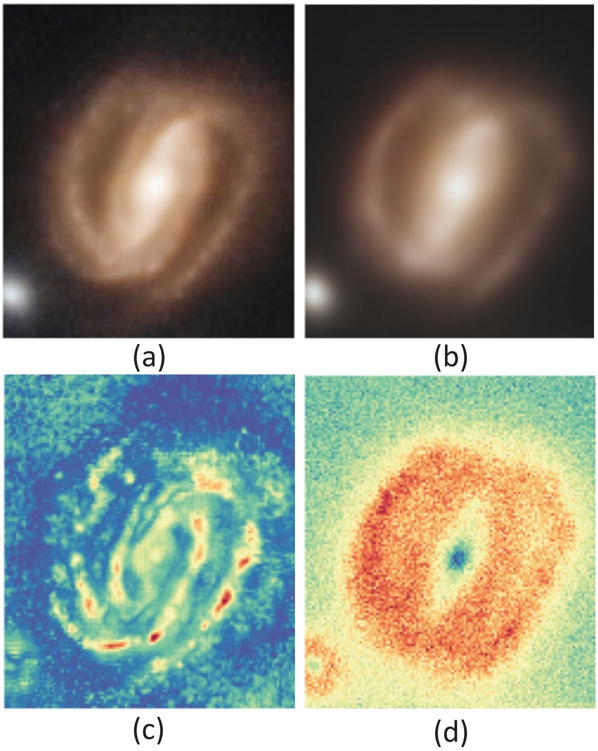


Figure 8. Uncertainty map for the galaxy: (**a**) target Webb image; (**b**) predicted Webb image; (**c**) true uncertainty with respect to the target image; (**d**) estimated uncertainty without the target image that reflects the variability in the generated images. The estimated PSUR: 28.99 dB.

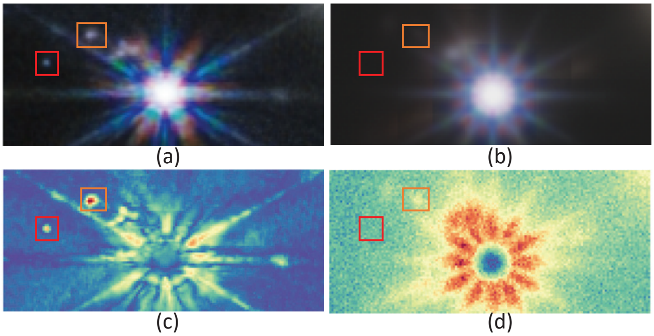


Figure 9. Uncertainty map for the star: (**a**) target Webb image; (**b**) predicted Webb image; (**c**) true uncertainty; (**d**) estimated uncertainty. The point sources, that were missed, and for which there is no sign in the uncertainty map, are highlighted with a red box. The point sources are missed in the predicted Webb, but there is a sign of one in the uncertainty map, which means it was present in some of the predictions. The estimated PSUR: 24.44 dB.

5. Implementation Details

We use PyTorch 1.12 [35] deep learning framework in all our experiments.

Data. We use crops from Hubble and Webb images of size 256 × 256 pixels in each experiment. All of the images used in training and validation are available at github.com/vkinakh/Hubble-meets-Webb, (accessed on 8 February 2024). We apply random horizontal and vertical flipping to each image pair of Hubble–Webb images as augmentation.

Pix2Pix and CycleGAN. In the experiments with Pix2Pix and CycleGAN, we use a convolutional architecture consisting of two convolutional layers for downsampling, nine residual blocks, and two transposed convolutional layers for upsampling for both the encoder and decoder. As discriminators, we use PatchGAN [7] with LSGAN loss [36], as provided in the original implementations. During training, we use an Adam [37] optimizer with a learning rate of 2×10^{-4} and a linear learning rate policy weight decay every 50 steps. Each model is trained for 100 epochs with a batch size of 64. For the experiments, we have used NVIDIA RTX 2080Ti GPU.

TURBO. In the experiments with TURBO [9], we use the same convolutional architectures for the encoder and decoder as in the Pix2Pix and CycleGAN experiments. TURBO consists of two convolutional generators: the first, $q_\phi(\mathbf{x}, \mathbf{z})$, generates Webb images from Hubble ones, and the second, $p_\theta(\mathbf{x}|\mathbf{z})$, generates Hubble images from Webb ones. We use four PatchGAN [7] discriminators: one for generated Webb samples $\mathcal{D}_{\mathbf{x}\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$, one for reconstructed Webb samples $\mathcal{D}_{\mathbf{x}\hat{\mathbf{x}}}(\hat{\mathbf{x}})$, one for generated Hubble images $\mathcal{D}_{\mathbf{z}\tilde{\mathbf{z}}}(\tilde{\mathbf{z}})$, and one for reconstructed Hubble images $\mathcal{D}_{\mathbf{z}\hat{\mathbf{z}}}(\hat{\mathbf{z}})$. Alternatively, the TURBO model can only use two discriminators: the first $\mathcal{D}_{\mathbf{z}}$ for generated and reconstructed Webb images, and the second $\mathcal{D}_{\mathbf{x}}$ for generated and reconstructed Hubble images. The results using two discriminators are shown in the ablation study in Table 1. As estimation and cycle losses, we use the ℓ_1-metric. We use the LSGAN discriminator loss [36], as in the Pix2Pix and CycleGAN experiments. Similarly, we use the Adam optimizer with a learning rate of 2×10^{-4} and a linear learning rate policy with decay every 50 steps. The model is trained for 100 epochs with a batch size of 64. For the experiments, we have used NVIDIA RTX 2080Ti GPU.

DDPM (Palette). In the experiments, we use a DDPM image-to-image translation model proposed in [11]. We use a UNet [38]-based noise estimator, with self-attention [39]. During training, we use a linear beta schedule with 2000 steps, 10^{-6} start, and 0.01 end. During inference, we use a DDPM scheduler with 1000 steps, 10^{-6} start, and 0.01 end. The model is trained for 1000 epochs with a batch size of 32. For the experiments, we have used NVIDIA A100 GPU.

During inference, since our images exceed 256 × 256 pixels, we employ a method known as *stride prediction* to predict patches of size 256 × 256 using a selected stride value. This method works systematically across the image: starting from the top-left corner at position $(0, 0)$, we predict the first patch, then move horizontally by stride s to predict the next, proceeding row by row until the entire image is covered. If the bottom or right edge is reached, the next row begins just below the starting point or back at the left edge, respectively. After predicting all patches, we save the images and track the prediction count for each pixel. The final pixel value is determined by averaging across all predictions for that pixel, ensuring a seamless image reconstruction.

Table 1. Ablation studies on paired models Pix2Pix and TURBO on locally synchronized data. All results are obtained on Galaxy Cluster SMACS 0723. The label "TURBO same D" corresponds to an approach, when the same discriminator is used for generated and reconstructed Webb and Hubble images. The label "LPIPS" denotes adding perceptual similarity loss.

Method	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	FID↓
$\mathcal{L}_1$	0.002	0.93	26.94	0.47	83.32
$\mathcal{L}_2$	0.002	0.93	26.98	0.47	76.03
$\mathcal{L}_1 + \mathcal{L}_2$	0.002	0.93	26.93	0.47	82.71
$\mathcal{L}_1$+ LPIPS	0.002	0.93	26.68	0.44	72.84
Pix2Pix	0.002	0.93	26.78	0.44	54.58
Pix2Pix + LPIPS	0.003	0.93	27.02	0.44	58.86
TURBO	0.003	0.92	25.88	0.41	43.36
TURBO + LPIPS	0.003	0.92	25.91	0.39	50.83
$\mathcal{L}^{\text{reverse}}$	0.002	0.93	26.15	0.45	70.51
$\mathcal{L}^{\text{reverse}}$ + LPIPS	0.002	0.93	26.13	0.46	67.52
TURBO same D	0.002	0.92	26.04	0.4	55.29
TURBO same D + LPIPS	0.002	0.92	26.13	0.39	55.88

6. Results

In this section, we report image-to-image translation results for the prediction of Webb telescope images based on Hubble telescope images. In Table 2, we report results for four setups: (a) unpaired setup; (b) paired setup with the synchronization with respect to celestial coordinates, where images were synchronized by hand; (c) paired setup with global synchronization, where the full image was synchronized using a single affine transform; and (d) paired setup with local synchronization, where the images were split into multiple patches and then each of the Hubble and Webb patches were synchronized individually. For each setup, we have defined a training set that covers approximately 80% of the input image of the galaxy clusters SMACS 0723, and the rest is used as a validation set for results. We make sure that the training and validation set cover different parts of the sky and never overlap even for a single pixel. When generating images for evaluation, since the validation images are larger than 256 × 256, we have used the stride prediction described above with a stride of $four$. It is shown in Table 2 that the synchronization of the data is very important, as all of the considered models perform best when the data are locally synchronized. This fact was not well addressed in previous studies, to the best of our knowledge. Also, we show that the DDPM-based image-to-image translation model outperforms the CycleGAN, Pix2Pix, and TURBO models in terms of MSE, SSIM, PSNR, FID and LPIPS metrics. The only downside of the DDPM model is its inference time, which is 1000 times longer than the inference time of Pix2Pix, CycleGAN and TURBO. This might be a serious limitation in practice, considering the size and number of astronomical images.

Table 2. Hubble-to-Webb results. All results are obtained on a validation set of Galaxy Cluster SMACS 0723.

Method	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	FID ↓
unpaired					
CycleGAN	0.010	0.83	20.11	0.48	128.12
paired: synchronization with respect to celestial coordinates					
Pix2Pix	0.007	0.87	21.37	0.5	102.61
TURBO	0.008	0.85	20.87	0.49	98.41
DDPM (Palette)	0.003	0.88	25.36	0.43	51.2
paired: global synchronization					
Pix2Pix	0.003	0.92	25.85	0.46	55.69
TURBO	0.003	0.91	25.08	0.45	48.57
DDPM (Palette)	0.002	0.94	28.12	0.45	43.97
paired: local synchronization					
Pix2Pix	0.002	0.93	26.78	0.44	54.58
TURBO	0.003	0.92	25.88	**0.41**	43.36
DDPM (Palette)	**0.001**	**0.95**	**29.12**	0.44	**30.08**

In Table 3, we compare parameter counts and inference times for a 256 × 256 image from the models considered in the study. The DDPM model is particularly noteworthy for its extensive parameter count, with both trainable and inference parameters reaching 62.641 Mio. It also necessitates 1000 generation steps, contributing to a longer inference time of approximately 42.77 seconds. Conversely, Pix2Pix, CycleGAN, and Turbo demonstrate a more streamlined parameter structure. These models employ generators with a uniform parameter count of 11.378 Mio and discriminators with 2.765 Mio parameters. Pix2Pix operates with one generator and one discriminator, CycleGAN with two of each, and Turbo with two generators and four discriminators. Despite the architectural differences, these models maintain compact trainable parameters, ranging from 14.143 Mio to 33.816 Mio, and achieve notably swift inference times, clocked at around 0.07 seconds. The inference time is averaged over 100 generations for each model on a single RTX 2080 Ti GPU with a batch size of one.

Table 3. Analysis of parameter complexity and inference time in image-to-image translation models.

Model	Trainable Params	Inference Paras	Inference Time
DDPM (1000 steps)	62.641 Mio	62.641 Mio	42.77 ± 0.18 s
Pix2Pix	14.143 Mio	11.378 Mio	0.07 ± 0.004 s
CycleGAN	28.286 Mio	11.378 Mio	0.07 ± 0.004 s
TURBO	33.816 Mio	11.378 Mio	0.07 ± 0.004 s

In Table 1, we perform ablation studies on the paired TURBO and Pix2Pix image-to-image translation models. We compare these models trained under various conditions: (a) with the $\mathcal{L}_1$ loss, which is the mean absolute error, (b) with the $\mathcal{L}_2$ loss, which is the mean squared error, (c) with both $\mathcal{L}_1$ and $\mathcal{L}_2$ losses, (d) with $\mathcal{L}_1$ loss and the Learned Perceptual Image Patch Similarity (LPIPS) loss using a VGG encoder [13]. We also explore Pix2Pix configurations, such as Pix2Pix with $\mathcal{L}_1$ loss plus a discriminator, Pix2Pix combined with LPIPS loss and a VGG encoder, along with variations of the TURBO model: TURBO with LPIPS loss, TURBO operating only in reverse pass, and TURBO using the same discriminator for both generated and reconstructed images. Models are trained and evaluated on data synchronized locally. As Table 1 indicates, Pix2Pix models and those without a discriminator perform better on paired metrics (MSE, PSNR, SSIM), whereas TURBO-based methods excel in image quality metrics (LPIPS, FID). Notably, the DDPM-based image-to-image translation method outperforms other methods discussed in the ablation study.

7. Conclusions

In this paper, we have proposed the use of image-to-image translation approaches for sensor-to-sensor translation in astrophysics for the task of predicting Webb images from Hubble. The novel TURBO framework serves as a versatile tool that outperforms existing GAN-based image-to-image translation methods, offering better quality in generated Webb telescope imagery and information-theoretic explainability. Furthermore, the application of DDPM for uncertainty estimation introduces a probabilistic dimension to image translation, providing a robust measure of reliability previously unexplored in this context. We show the importance of synchronization in paired image-to-image translation approaches.

This research not only paves the way for improved astronomical observations by leveraging advanced computational techniques but also advocates for the application of these methods in other domains where image translation and uncertainty estimation are crucial. As we continue to venture into the cosmos, the methodologies refined here will undoubtedly become instrumental in interpreting and maximizing the utility of the data we collect from advanced telescopes.

8. Future Work

Out future research will include an approach to refine and enhance the methodologies discussed in this paper. A particular focus will be directed towards improving the TURBO model, which, while being computationally efficient, currently lags behind DDPM in terms of performance. TURBO model improvement will be mostly focused on architectural improvements of generators. In parallel, we plan to undertake a thorough investigation into the resilience of our applied methods against various data preprocessing techniques, including different forms of interpolation. This study aims to ensure the robustness and adaptability of our models across a spectrum of data manipulation scenarios. Moreover, the exploration of existing sampling techniques within DDPMs will be pursued with the goal of expediting inference times. This focus is expected to significantly improve the models' efficiency, rendering them more suitable for real-time applications.

The current research specifically focuses on the analysis of RGB pseudocolor images. A significant portion of our future work will be dedicated to the meticulous training and evaluation of the proposed models on raw astrophysical data. This will involve the integration of specialized astrophysical metrics designed to align with the unique properties of such data, thereby assuring that our models are not only statistically sound but also truly resonate with the practical demands and intricacies of astrophysical research. We aspire to bridge the gap between theoretical robustness and real-world applicability, setting the stage for transformative developments in the field of image-to-image translation in astrophysical data analysis.

Author Contributions: Conceptualization, S.V. and V.K.; methodology, S.V., V.K., and G.Q.; software, V.K. and Y.B.; validation, S.V. and M.D.; formal analysis, S.V. and G.Q.; investigation, V.K.; resources, V.K.; data curation, V.K.; writing—original draft preparation: V.K., S.V., G.Q. and Y.B.; writing—review and editing, V.K., Y.B., G.Q., M.D., T.H., D.S. and S.V.; visualization, V.K., S.V. and G.Q.; supervision, S.V., T.H. and D.S.; project administration, S.V., T.H. and D.S.; funding acquisition, S.V. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the SNF Sinergia project (CRSII5-193716) Robust deep density models for high-energy particle physics and solar flare analysis (RODEM).

Data Availability Statement: The code and data used in the study can be accessed at public repository: github.com/vkinakh/Hubble-meets-Webb, (accessed on 8 February 2024). The experimental results can be accessed at hubble-to-webb.herokuapp.com, (accessed on 8 February 2024).

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

Abbreviations

The following abbreviations are used in this manuscript:

GAN	Generative adversarial network
DDPM	Denoising diffusion probabilistic model
MSE	Mean squared error
PSNR	Peak signal to noise ratio
LPIPS	Learned perceptual image patch similarity
FID	Fréchet inception distance
RGB	Red, green, blue
AE	Auto-encoder
SSIM	Structural similarity index measure
TURBO	Two-way Uni-directional Representations by Bounded Optimisation
HST	Hubble Space Telescope
JWST	James Webb Space Telescope
LSGAN	Least Squares Generative Adversarial Network
SIFT	Scale-Invariant Feature Transform
RANSAC	Random Sample Consensus
ESA	European Space Agency
NASA	National Aeronautics and Space Administration
STScI	Space Telescope Science Institute

References

1. Garner, J.P.; Mather, J.C.; Clampin, M.; Doyon, R.; Greenhouse, M.A.; Hammel, H.B.; Hutchings, J.B.; Jakobsen, P.; Lilly, S.J.; Long, K.S.; et al. The James Webb space telescope. *Space Sci. Rev.* **2006**, *123*, 485–606. [CrossRef]
2. Lallo, M.D. Experience with the Hubble Space Telescope: 20 years of an archetype. *Opt. Eng.* **2012**, *51*, 011011. [CrossRef]
3. Lin, Q.; Fouchez, D.; Pasquet, J. Galaxy Image Translation with Semi-supervised Noise-reconstructed Generative Adversarial Networks. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5634–5641.
4. Schaurecker, D.; Li, Y.; Tinker, J.; Ho, S.; Refregier, A. Super-resolving Dark Matter Halos using Generative Deep Learning. *arXiv* **2021**, arXiv:2111.06393.
5. Racca, G.D.; Laureijs, R.; Stagnaro, L.; Salvignol, J.C.; Alvarez, J.L.; Criado, G.S.; Venancio, L.G.; Short, A.; Strada, P.; Bönke, T.; et al. The Euclid mission design. In Proceedings of the Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave, Edinburgh, UK, 19 July 2016; Volume 9904, pp. 235–257.
6. Hall, P.; Schillizzi, R.; Dewdney, P.; Lazio, J. The square kilometer array (SKA) radio telescope: Progress and technical directions. *Int. Union Radio Sci. URSI* **2008**, *236*, 4–19.
7. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
8. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
9. Quétant, G.; Belousov, Y.; Kinakh, V.; Voloshynovskiy, S. TURBO: The Swiss Knife of Auto-Encoders. *Entropy* **2023**, *25*, 1471. [CrossRef] [PubMed]
10. Dhariwal, P.; Nichol, A. Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794.
11. Saharia, C.; Chan, W.; Chang, H.; Lee, C.; Ho, J.; Salimans, T.; Fleet, D.; Norouzi, M. Palette: Image-to-image diffusion models. In Proceedings of the ACM SIGGRAPH 2022 Conference Proceedings, Vancouver, BC, Canada, 7–11 August 2022; pp. 1–10.
12. 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] [PubMed]
13. 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 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
14. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf (accessed on 3 January 2024).
15. NASA. Webb vs Hubble Telescope. Available online: https://www.jwst.nasa.gov/content/about/comparisonWebbVsHubble.html (accessed on 6 January 2024).
16. Science, N. Hubble vs. Webb. Available online: https://science.nasa.gov/science-red/s3fs-public/atoms/files/HSF-Hubble-vs-Webb-v3.pdf (accessed on 6 January 2024).
17. Space Telescope Science Institute. Webb Space Telescope. Available online: https://webbtelescope.org (accessed on 6 January 2024).

18. European Space Agency. Hubble Space Telescope. Available online: https://esahubble.org (accessed on 6 January 2024).
19. Pang, Y.; Lin, J.; Qin, T.; Chen, Z. Image-to-image translation: Methods and applications. *IEEE Trans. Multimed.* **2021**, *24*, 3859–3881. [CrossRef]
20. Liu, M.Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; Kautz, J. Few-shot unsupervised image-to-image translation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 10551–10560.
21. Zhang, R.; Isola, P.; Efros, A.A. Colorful image colorization. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 649–666.
22. Hui, Z.; Gao, X.; Yang, Y.; Wang, X. Lightweight image super-resolution with information multi-distillation network. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 2024–2032.
23. Patel, D.; Patel, S.; Patel, M. Application of Image-To-Image Translation in Improving Pedestrian Detection. In *Artificial Intelligence and Sustainable Computing*; Pandit, M., Gaur, M.K., Kumar, S., Eds.; Springer Nature: Singapore, 2023; pp. 471–482.
24. Kaji, S.; Kida, S. Overview of image-to-image translation by use of deep neural networks: Denoising, super-resolution, modality conversion, and reconstruction in medical imaging. *Radiol. Phys. Technol.* **2019**, *12*, 235–248. [CrossRef]
25. Liu, M.-Y.; Breuel, T.; Kautz, J. Unsupervised image-to-image translation networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf (accessed on 3 February 2024).
26. Tripathy, S.; Kannala, J.; Rahtu, E. Learning image-to-image translation using paired and unpaired training samples. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 51–66.
27. Vojtekova, A.; Lieu, M.; Valtchanov, I.; Altieri, B.; Old, L.; Chen, Q.; Hroch, F. Learning to denoise astronomical images with U-nets. *Mon. Not. R. Astron. Soc.* **2021**, *503*, 3204–3215. [CrossRef]
28. Liu, T.; Quan, Y.; Su, Y.; Guo, Y.; Liu, S.; Ji, H.; Hao, Q.; Gao, Y. Denoising Astronomical Images with an Unsupervised Deep Learning Based Method. *arXiv* **2023**, . [CrossRef]
29. NASA/IPAC. Galaxy Cluster SMACS J0723.3-7327. Available online: http://ned.ipac.caltech.edu/cgi-bin/objsearch?search_type=Obj_id&objid=189224010 (accessed on 6 January 2024).
30. Bohn, T.; Inami, H.; Diaz-Santos, T.; Armus, L.; Linden, S.T.; Surace, J.; Larson, K.L.; Evans, A.S.; Hoshioka, S.; Lai, T.; et al. GOALS-JWST: NIRCam and MIRI Imaging of the Circumnuclear Starburst Ring in NGC 7469. *arXiv* **2022**, arXiv:2209.04466.
31. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
32. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
33. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
34. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
35. 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 *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: 2019; Volume 32. Available online: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf (accessed 3 February 2024).
36. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2794–2802.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: 2017; Volume 30. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (accessed 3 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

Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model

Talal H. Noor *, Ayman Noor, Ahmed F. Alharbi, Ahmed Faisal, Rakan Alrashidi, Ahmed S. Alsaedi, Ghada Alharbi, Tawfeeq Alsanoosy and Abdullah Alsaeedi

Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia; anoor@taibahu.edu.sa (A.N.); tu4101709@taibahu.edu.sa (A.F.A.); tu4107228@taibahu.edu.sa (A.F.); tu4102973@taibahu.edu.sa (R.A.); tu4101162@taibahu.edu.sa (A.S.A.); gfharbi@taibahu.edu.sa (G.A.); tsanoosy@taibahu.edu.sa (T.A.); aasaeedi@taibahu.edu.sa (A.A.)

* Correspondence: tnoor@taibahu.edu.sa

Abstract: Sign language is an essential means of communication for individuals with hearing disabilities. However, there is a significant shortage of sign language interpreters in some languages, especially in Saudi Arabia. This shortage results in a large proportion of the hearing-impaired population being deprived of services, especially in public places. This paper aims to address this gap in accessibility by leveraging technology to develop systems capable of recognizing Arabic Sign Language (ArSL) using deep learning techniques. In this paper, we propose a hybrid model to capture the spatio-temporal aspects of sign language (i.e., letters and words). The hybrid model consists of a Convolutional Neural Network (CNN) classifier to extract spatial features from sign language data and a Long Short-Term Memory (LSTM) classifier to extract spatial and temporal characteristics to handle sequential data (i.e., hand movements). To demonstrate the feasibility of our proposed hybrid model, we created a dataset of 20 different words, resulting in 4000 images for ArSL: 10 static gesture words and 500 videos for 10 dynamic gesture words. Our proposed hybrid model demonstrates promising performance, with the CNN and LSTM classifiers achieving accuracy rates of 94.40% and 82.70%, respectively. These results indicate that our approach can significantly enhance communication accessibility for the hearing-impaired community in Saudi Arabia. Thus, this paper represents a major step toward promoting inclusivity and improving the quality of life for the hearing impaired.

Keywords: deep learning; real-time detection; Arabic sign language recognition; CNNs; LSTM

Citation: Noor, T.H.; Noor, A.; Alharbi, A.F.; Faisal, A.; Alrashidi, R.; Alsaedi, A.S.; Alharbi, G.; Alsanoosy, T.; Alsaeedi, A. Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model. *Sensors* **2024**, *24*, 3683. https://doi.org/10.3390/s24113683

Academic Editors: Sukho Lee and Dae-Ki Kang

Received: 24 April 2024
Revised: 31 May 2024
Accepted: 3 June 2024
Published: 6 June 2024

1. Introduction

Sign language is a communication method utilized by deaf individuals and encompasses a series of hand gestures and symbols [1]. It is also employed by hearing individuals to facilitate communication with the deaf community. Predominantly, sign language is concept-based, with each gesture or symbol representing a distinct idea or concept. It uses four major manual components that comprise (1) finger configuration, (2) hand movement, (3) hand orientation, and (4) hand location relative to the body [2,3]. Compared to other gestures, sign language is the most structured. On the one hand, it has a large set of signs where each sign has a specific meaning [4]. On the other hand, this contrasts with word-based communication systems. Nonetheless, certain words and names lack direct equivalents in sign language. To address this, the deaf community often resorts to using a hand (i.e., finger) alphabet to spell out such words, ensuring clarity and precision in communication. This approach highlights the adaptability and inclusivity of sign language as a communication tool [5].

The Kingdom of Saudi Arabia is home to a sizable deaf population of about 229,541, many of whom are not provided with appropriate care in public venues because of a lack of interpreters, as the General Authority for Statistics has shown in recent years [6]. According to the Center for Strategic and International Studies (CSIS) [7], in the state of California, the ratio of sign language interpreters to hearing-impaired individuals is approximately 1:46. This indicates a relatively high availability of interpreters for the deaf community. In contrast, Saudi Arabia presents a starkly different scenario, with the ratio being approximately 1:93,000. This vast disparity highlights significant differences in the availability of sign language interpretation services between the two regions, underscoring a potential area of concern in terms of accessibility and support for the hearing-impaired population in Saudi Arabia. Moreover, most current research utilizes either only letters/alphabets to translate Arabic Sign Language (ArSL) or sensors to facilitate this process [4,8,9]. Thus, there is a need to conduct further research on ArSL [9,10].

In addressing the shortage of sign language interpreters, the role of technology, particularly machine-based communication methods, becomes crucial in bridging this gap. Advances in machine learning have led to the development of automated sign language translation systems. These systems employ sophisticated algorithms and gesture recognition technologies to translate sign language into spoken and written language, and vice versa. Such technological solutions offer a promising path to alleviate interpreter scarcity, particularly in areas such as the Kingdom of Saudi Arabia, where the ratio of interpreters to hearing-impaired individuals is exceedingly low. Machine-based communication can provide real-time, on-demand translation services, making communication more accessible and inclusive for the deaf community.

Furthermore, contemporary technological approaches, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), are instrumental in enhancing the efficacy of automated ASL translation systems. CNNs can process and interpret visual information, making them highly suitable for recognizing and analyzing the intricate hand gestures and facial expressions inherent in sign language [11–14]. LSTM networks, a form of recurrent neural network, excel in handling sequential data, thus effectively capturing the dynamic and temporal aspects of sign language [13–16]. Hence, in this paper, both CNNs and LSTMs were employed to develop the proposed model.

The novelty and main contribution of this paper focus on tackling a critical societal issue: the lack of sign language interpreters in the Kingdom of Saudi Arabia, which has left a significant hearing-impaired population without adequate support in public spaces. This research highlights the importance and versatility of Arabic Sign Language (ArSL) as a communication tool and leverages technology such as artificial intelligence to improve accessibility for the deaf community in Saudi Arabia. The key contributions of this paper include:

- Identifying the Accessibility Gap: This research comprehensively analyzes the accessibility gap faced by the hearing-impaired population in Saudi Arabia compared to regions with more interpreters such as California, USA. By highlighting the stark contrast in interpreter ratios, this paper sheds light on a crucial issue that requires attention and action.
- Leveraging Deep Learning: We propose a novel hybrid model to capture the spatio-temporal aspects of sign language. The hybrid model consists of a CNN classifier to extract spatial features from sign language data and an LSTM classifier to extract spatial and temporal characteristics to handle sequential data. This hybrid model facilitates the process of automatic sign language recognition in real time from either spoken or written language, addressing the shortage of qualified interpreters.
- Building a Custom Annotated Dataset: We create a dataset of 20 different universal words, resulting in 4000 images for ArSL. It consists of 10 static gesture words and 500 videos for 10 dynamic gesture words. The dataset can be used to train and evaluate ASL recognition models.
- Validation: We conduct a comprehensive evaluation of the hybrid model on our dataset and compare its performance on 20 different words (i.e., 10 static gesture words and 10 dynamic gesture words).

The remainder of this paper is organized as follows. The related works are covered in Section 2. Section 3 reviews the system architecture and the hybrid model. Section 4 describes the proposed model's implementation, and Section 5 discusses the experimental results. Section 6 presents the concluding remarks and discusses future work.

2. Related Works

The issue of sign language recognition has recently attracted the attention of many researchers. Several survey articles have been specifically written to draw attention to the problems in the field of sign language recognition [1,10,17–19]. Some researchers have focused on ArSL recognition. For example, in [20], the researchers developed a real-time system for ArSL recognition using You Only Look Once (YOLO), in particular, the YOLOv5 model. The process began with a dataset of 5600 images of 28 ArSL signs, which was expanded to 15,088 images through augmentation techniques, including resizing, normalization, Gaussian blur, noise addition, affine transformations, and grayscale conversion. Three versions of the YOLOv5 model (small, medium, and large) were trained and evaluated, with the YOLOv5-large version achieving the best performance, with 99.5% in precision, 99.4% in recall, and 99.4% in mean average precision (mAP@.5). The YOLOv5s model, however, was identified as the most suitable for real-time recognition due to its high speed, with an average of 121 FPS and satisfactory mAP results.

In [21], the authors developed an Alphabet Recognition System for ArSL using a faster Region-based Convolutional Neural Network (RCNN). They collected a non-depth camera dataset of 15,360 images, containing hand movements against different backgrounds, captured using standard phone cameras to evaluate the system. The data were divided in such a way that 60% was used for training. Meanwhile, 20% of the total image collection was allocated for 3-fold cross-validation. The remaining 20% was set aside for testing. The proposed approach achieved 93% accuracy.

In [22], the authors developed a recognition system for Arabic, specifically Egyptian Sign Language (ESL), to improve communication with people with hearing impairment. It was implemented in a video-based system to serve the local deaf community in Egypt, utilizing two different neural network architectures: the first was a Convolutional Neural Network (CNN) to extract spatial features, and the second consisted of a CNN followed by Long Short-Term Memory (LSTM) to extract spatio-temporal features. The authors created a dataset by capturing videos under different lighting conditions, angles, and camera positions. The dataset consisted of nine classes, with videos recorded by two signers at five different locations. Each location contributed 20 videos, resulting in a total of 100 videos per chapter. Thus, the entire dataset included 900 videos. The researchers evaluated the performance of the two architectures. Using only the CNN, they achieved up to 90% accuracy. However, when they combined the CNN with LSTM, the accuracy dropped to 72%.

In [23], the authors developed a system to recognize ArSL letters. They used a CNN classifier to extract the exact position of the hand. The authors used an external dataset containing 1160 images of 416 × 416 pixels and merged it with a dataset they created containing 28 classes, where each category contained about 135 images describing an Arabic letter. The dataset was split, with 60% of the data used for training, 20% for testing, and 20% for validation. Their system achieved an accuracy of 97.07%.

Unlike previous research studies that focus solely on either image-based or video-based ArSL recognition, in this paper, we propose a hybrid model that can recognize ArSL in both forms. Furthermore, unlike previous research works that train their models on the 28 Arabic alphabet letters or only train their models on 9 Arabic words, our hybrid model is trained on 20 different Arabic words for videos and images using our dataset containing 4000 images and 500 videos.

Some researchers have focused on the recognition of other sign languages, such as English, Indian, Japanese, Bangla, Indonesian, Korean, Indian (Assamese), and Turkish. For example, [24] focuses on American-English Sign Language Translation (ASLAT), where various datasets and YOLO network models were utilized to enhance the accuracy and efficiency of sign language recognition. The open-access ASL alphabet dataset was partially used, with 24,000 images out of a total of 87,000, covering 26 ASL alphabet letters, excluding "J" and "Z", and included additional signs for SPACE, DELETE, and NOTHING, though specific accuracy figures for this dataset were not mentioned. Another key component was the ASLYset dataset, comprising 5200 images and featuring 24 ASL alphabet signs (also excluding "J" and "Z") and two extra signs, "SP" and "FN". The testing dataset contained 1300 images with 50 images per sign, excluding "J" and "Z". In terms of YOLO network models, two architectures, namely A1 and A2, were tested (A1 based on YOLOv3-tiny and A2 based on YOLOv3) with input image sizes of 288 × 288, 352 × 352, and 416 × 416 pixels. The highest mAP achieved by one of the YOLO models was 81.76%, although specific accuracy details for each model were not provided. The Bidirectional LSTM for Spelling Correction trained on a dataset based on 235 English sentences and expanded to 11,750 training sentences through noisy word generation showed a training accuracy of 98.07%.

In [25], the research aimed at classifying Indian Sign Language gestures, and an image database was created with images of hands of varying sizes and complexions. These images, originally sized at 768 × 1024, were pre-processed and resized to 256 × 256 to facilitate dimensionality reduction, which is crucial for faster classification. Principal Component Analysis (PCA) was employed for this purpose, not only reducing the time and storage space required but also improving the performance of the machine learning model by removing multi-collinearity and making the data easier to visualize. The study utilized two machine learning techniques to train the system: the K-Nearest Neighbor (KNN) algorithm and the backpropagation algorithm. The dataset comprised 220 images of double-handed and 800 images of single-handed Indian Sign Language alphabets, representing English letters and numbers. Using the KNN technique with K = 1, a 100% pattern recognition rate was accomplished, whereas the backpropagation technique yielded a recognition rate between 94 and 96%.

In [26], a sophisticated system for Japanese Sign Language (JSL) recognition was introduced, leveraging the Microsoft Kinect v2 sensor to capture sign motion feature parameters. The system utilizes a JSL dictionary, which encompasses a comprehensive vocabulary of approximately 2600 signs to categorize a variety of hand poses. An integral component of the system is its adoption of a contour-based technique for hand pose recognition, innovated by Keogh, which is particularly adept at handling situations where fingers are partially occluded. The experimental phase of the research involved the analysis of 892 samples, derived from 223 distinct signs, each performed by two interpreters and repeated twice, although the number of images was unspecified. These samples were instrumental in capturing the nuances of sign motion and hand positions. The findings from these experiments highlighted an improvement in recognition rates, with an increase in the number of wedges utilized. Despite a recognition accuracy of 33.8% for signs not previously included in the training data, the paper did not detail the dataset sizes for the employed models, such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs), which were used for recognizing hand movement and position. This research marks a significant step forward in the domain of sign language recognition by presenting a real-time system that seamlessly integrates the elements of motion, position, and pose for JSL recognition.

In [27], the authors proposed a hand gesture recognition system, where a webcam was employed to collect a dataset comprising 2000 images of American-English Sign Language (ASL) gestures, specifically focusing on 10 static alphabet signs (A, B, C, D, K, N, O, T, Y), captured under various lighting conditions to ensure robustness. The system leverages advanced image processing techniques, including HSV color extraction for effective background removal, alongside segmentation and morphological operations to refine the quality of the images. These pre-processed images are then fed into a Convolutional Neural Network (CNN), which is trained on 1600 of these images, with the remaining 400 images reserved for testing, adhering to an 80:20 training-to-testing ratio. The result is a highly accurate gesture recognition system, evidenced by a remarkable 100% prediction accuracy rate achieved by the CNN model, showcasing its capability to accurately identify gestures from images. This system not only demonstrates the potential of CNNs in image-based recognition tasks but also highlights the importance of comprehensive data collection and meticulous pre-processing in developing effective recognition algorithms. The dataset comprised only 2000 photos, averaging 200 images for each gesture.

In [28], the researchers analyzed and interpreted Indian Sign Language using LSTM networks, employing a dataset representing 40 different actions, with 30 videos for each action, each containing 30 frames, ensuring diversity in gender and signing capabilities. The videos were recorded using high-resolution cameras in controlled environments to guarantee the quality and realistic representation of the data. The data underwent multiple stages, including video processing and feature extraction, utilizing techniques such as Principal Component Analysis (PCA) and wavelet analysis, paving the way for the application of LSTM networks to learn the temporal patterns of the signs. Despite challenges and limitations related to the dataset's size and the full representation of sign language diversity, the study achieved a training accuracy of 87%, indicating the model's ability to accurately classify the signs. In [29], the authors proposed an approach to train and detect Bengali Sign Language using deep learning. The authors used a CNN classifier to train each sign individually. The system was trained on samples containing different signs used in Bengali Sign Language. They captured their dataset using an Intel RealSense webcam. They labeled 90 classes in total, and each class represented one label. The data were split into 80% for training and the remaining 20% for testing and validation. The accuracy of their model reached 78%.

In [30], the authors proposed a YOLOv3 method for Indonesian Sign Language recognition. The authors developed a system that can automatically translate sign language into text to facilitate communication between deaf individuals and those who do not understand sign language. The system was designed to process video data inputs in real time using an object detection method, such as YOLOv3, based on a CNN classifier. The dataset used in this research was collected and captured independently by the authors under different conditions and based on Indonesian Sign Language, and it consisted of images and videos representing 24 categories of sign language gestures. The dataset contained approximately 160 to 220 pictures, representing the letters A to Z, excluding J and R. When using image data, the data were split into 80% for training and 20% for testing. Their system achieved 100% accuracy. When using video data, the system achieved 72.97% accuracy.

In [31], the authors developed a video data input processing system for Korean Sign Language recognition by using human key points extracted from the face, hand, and body parts as input for a recurrent neural network (RNN). They presented a dataset consisting of 10,480 high-resolution, high-quality video clips. They trained and tested their model using a dataset of 1000 videos, each recording 100 different sign language sentences. The system achieved a classification accuracy of 89.5% for 100 sentences that could be used in emergencies. This suggests that the system can be particularly useful in critical situations where clear communication is essential.

To overcome the difficulties in sign language recognition, the authors of [32] proposed an end-to-end skeleton-based multi-feature multi-stream multi-level information sharing network (TMS-Net). Combining joint information with global features, bone features with local features, and angle features with scale invariance allows TMS-Net to improve input richness. Comprehensive extraction and exploitation of skeleton feature information are ensured by its multi-stream structure and multi-level information-sharing mechanism. Based on a single modality input, the experimental results demonstrated that TMS-Net outperformed state-of-the-art approaches on the WLASL-2000 (56.4%), AUTSL (96.62%), and MSASL (65.13%) datasets. Furthermore, the practical efficiency of TMS-Net was demonstrated in an SLR-based Human–Robot Interaction (HRI) experiment, underscoring its potential for real-world applications.

In [33], the authors proposed the Natural Language-Assisted Sign Language Recognition (NLA-SLR) framework as a solution to the issue of visually indistinguishable signs (VISigns) in sign languages. They introduced language-aware label smoothing, which generates soft labels with smoothing weights based on normalized semantic similarities among glosses, to improve training for VISigns with similar semantic meanings. They provided an inter-modality mix-up approach that combines vision and gloss data for VISigns with different semantic meanings, improving separability under blended label supervision. Moreover, a unique backbone, the video-keypoint network, extracts knowledge from sign videos across various temporal receptive fields by modeling both RGB films and human body key points. The empirical findings revealed that NLA-SLR is effective in enhancing the accuracy of sign language recognition, achieving state-of-the-art performance on the MSASL, WLASL-2000, and NMFs-CSL datasets, achieving an average accuracy of 61.26% on the WLASL-2000 dataset. Unlike previous research that proposed NLA-SLR, achieving 61.26% accuracy, this study offers a hybrid approach for Arabic Sign Language (ArSL), achieving an average accuracy of 88.55% for both sub-models.

In [34], the authors presented two deep neural network-based model designs to address the challenges in gesture recognition and Audio-Visual Speech Recognition (AVSR). The main innovations in AVSR are the end-to-end model that utilizes three modality fusion approaches—prediction-level, feature-level, and model-level—as well as the methodologies for fine-tuning both visual and audio features. The deep neural network-based model consists of two architectures: (i) a visual model architecture that leverages Two-Dimensional Convolutional Neural Networks and Bidirectional Long Short-Term Memory (2DCNN+BiLSTM), Three-Dimensional Convolutional Neural Networks (3DCNN), or Three-Dimensional Convolutional Neural Networks and Bidirectional Long Short-Term Memory (3DCNN+BiLSTM), and (ii) an audio model architecture that leverages Residual Networks (ResNets), Visual Geometry Group (VGG), or Pretrained Audio Neural Networks (PANNs). The novel aspect of gesture identification lies in a distinct set of spatio-temporal features (STF), which take lip articulation information into account. The experimental results showed the performance of the models on two large-scale corpora, obtaining an AVSR accuracy of 98.76% on the LRW dataset and a gesture recognition rate of 98.56% on the AUTSL dataset, despite the lack of combined task datasets. Unlike previous research that focused on Audio-Visual Speech Recognition (AVSR), this study focuses on Arabic Sign Language (ArSL).

In [35], the authors focused on the issue of conventional machine learning techniques requiring significant data collection and classification. They presented a novel deep learning architecture, namely 3D-CLDNN, that makes use of sEMG signals and depth vision. By automatically classifying depth data using a self-organizing map and predicting gestures using only sEMG data, it simplifies the procedure. With its 84.4% accuracy rate and fast processing time, the approach is appropriate for real-time applications involving human–machine interaction.

In [36], the authors developed a method for recognizing Indian (Assamese) Sign Language. They prepared a dataset of 2D and 3D images of Assamese gestures containing nine Assamese alphabets captured using a Microsoft Kinect sensor and an RGB webcam. They used the MediaPipe framework to detect landmarks in images and trained them using a feedforward neural network. Their model achieved up to 99% accuracy. However, there was no mention of the size of the dataset used. Unlike previous research works that focus on the recognition of other sign languages such as English, Indian, Japanese, Bangla, etc., in this paper, we focus on the recognition of Arabic Sign Language (ArSL). Moreover, unlike previous research works that focused on either image-based or video-based sign language recognition, in this paper, we propose a hybrid model that can recognize ArSL in both forms.

In [37], the authors addressed the issue of clearly delineating boundaries for sign words in systems for Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). They observed that traditional methods based on pre-trained models are less flexible and computationally demanding, but intermediate gloss prediction greatly improves SLT performance. In response to these problems, they suggested the Sign2Pose gloss prediction transformer, which lowers processing overhead and increases accuracy by using manually developed posture feature extraction techniques. To effectively detect key frames in sign movies, the authors used the Euclidean distance technique and a modified version of the HD algorithm. By including YOLOv3 to accurately detect hand movements, their model outperformed current pose-based techniques by 15–20% in accuracy. Their model achieved a new standard for accuracy and efficiency by outperforming other methods on the word-level ASL data corpus. Unlike previous research that proposed the Sign2Pose gloss prediction transformer model, which scored 80.90% in accuracy, this study offers a hybrid approach for Arabic Sign Language (ArSL), achieving an average accuracy of 88.55% for both sub-models.

In [38], the authors concentrated on the issue of large vocabulary in practical contexts for recognizing sign language. The authors presented a large-scale multi-modal Turkish Sign Language dataset, namely Ankara University Turkish Sign Language (AUTSL), which included baseline and benchmark models for evaluating performance. The dataset consisted of 226 signs executed by 43 signers, for a total of 38,336 isolated sign video clips. Microsoft Kinect v2 was used to record the RGB, depth, and skeleton modalities for each sample. The authors trained multiple deep learning models using CNNs for feature extraction and both unidirectional and bidirectional LSTMs to capture temporal information, which was then improved by feature-pooling modules and temporal attention. They also created benchmark training and testing sets for user-independent evaluations. The reported results on the AUTSL and Montalbano datasets were competitive, with accuracies of 96.11% and 95.95%, respectively. The limitations inherent in the user-independent benchmark dataset were highlighted, with the best baseline model achieving an accuracy of 62.02%. The authors focused on recognizing Turkish Sign Language, whereas our paper concentrates on the recognition of Arabic Sign Language (ArSL).

3. System Architecture and the Hybrid Model

To overcome the lack of Arabic interpreters for people with hearing impairment, especially in Saudi Arabia, an architecture for sign language recognition is presented in this work that enables automatic real-time translation between sign language and spoken or written language. The architecture is shown in Figure 1 and consists of four layers: the data acquisition layer, the mobile network layer, the cloud layer, and the sign language recognition layer.

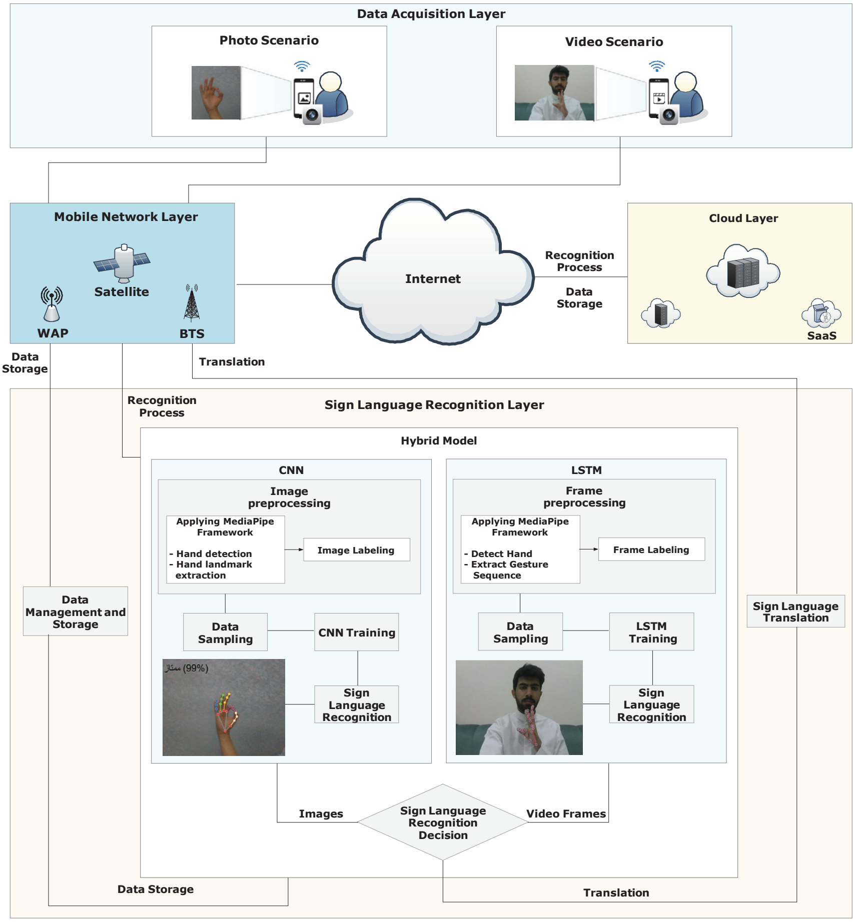


Figure 1. Real-time ArSL system architecture.

3.1. Architecture Layers

Each layer of the architecture is responsible for a set of tasks and interacts with the other layers. The architecture uses a streamlined pipeline design to make communication accessible and intuitive. The architecture layers are as follows:

1. *Data Acquisition Layer:* This layer gathers the sign language data that require translation using images collected from cameras (e.g., webcams, smartphone cameras, wearable gadget cameras, etc.) or videos collected from cameras, which are broken into frames. The images and frames are then sent through the *mobile network layer* and the *sign language recognition layer* for processing.

2. *Mobile Network Layer:* This layer connects the *data acquisition layer* and the *sign language recognition layer*. It is made up of several Wireless Access Points (WAPs), Base Transceiver Stations (BTSs), and satellites to enable communication. The information that is provided includes the ID of the camera and the images or video frames that contain the hand landmark or gesture sequence that needs to be translated. This layer makes communication easy and straightforward through a streamlined pipeline design.

3. Cloud Layer: For the other layers, this layer offers Infrastructure as a Service (IaaS) and Software as a Service (SaaS), making it possible for data to be stored and shared across layers via the Internet. It also gives the system security, scalability, and dependability. By employing a hybrid model for processing the deep learning models, including the training and learning phases, we use SaaS, a cloud computing model that does not require management of the underlying infrastructure. The management and storage of data, including training and recognition data for upcoming training, are achieved using the IaaS cloud computing architecture. With this computing approach, resources like servers, storage, and networks can grow as needed without needing to be managed.

4. Sign Language Recognition Layer: This layer utilizes the proposed *hybrid model* on each image or video frame that comes from the *data acquisition layer*, which contains a hand landmark or gesture sequence to be translated. The *hybrid model* consists of two deep learning models, namely Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), each of which consists of several modules:

- **Image/Frame Pre-processing:** This module employs the Google MediaPipe framework (https://developers.google.com/mediapipe/framework, accessed on 23 February 2024) to extract hand landmarks or gesture sequences from a set of images or frames. The images or frames are organized into separate directories, each representing a unique category. For every image or frame, the coordinates of detected hand landmarks or gesture sequences are captured and flattened into a list. These lists are then compiled into a 'data' list while corresponding category labels are added to the 'labels' list. The entire dataset, consisting of hand landmarks or gesture sequences and their associated labels, is stored in a pickle file (i.e., in the *cloud layer*).
- **Data Sampling:** This module is a crucial aspect of deep learning, particularly when dealing with data that require training and testing for their models. It randomly splits the data into 80% for training and 20% for testing. This division ensures that both models (i.e., CNN and LSTM in the hybrid model) learn patterns and relationships from a diverse range of examples during the training phase while also assessing their performance on unseen data during testing. The training set, comprising 80% of the data, is used to train the models, which allows them to learn from a large variety of hand gestures and their corresponding landmarks. On the other hand, the remaining 20% of the data reserved for testing serves as an independent validation set to evaluate both models' performance and generalization ability on new, unseen examples.
- **CNN Training:** The CNN sub-model of our *hybrid model* is designed to detect human hands by leveraging the Google MediaPipe framework to identify the hand's 21 3D landmarks. These landmarks are crucial for understanding hand gestures and movements. To train the CNN sub-model, we utilize pre-processed data stored in a pickle file (i.e., in the *cloud layer*), where the hand landmarks and corresponding labels are organized. We then employ the Random Forest Classifier (RFC) to predict the output of these landmarks. The RFC is an ensemble learning algorithm that constructs multiple decision trees during training. It outputs the mode of the classes (i.e., classification) or the mean prediction (i.e., regression) of the individual trees. Each decision tree in the ensemble is trained on a random subset of the training data. During prediction, each tree contributes a decision, with the final output determined by a majority or averaging mechanism. This combined approach helps the CNN sub-model to accurately detect and interpret human hand gestures in real-time applications.
- **LSTM Training:** The LSTM sub-model of our *hybrid model* uses an approach that focuses on capturing sequential patterns in time-series data, particularly in the context of hand gesture sequences. To train the LSTM sub-model, we utilize pre-processed data stored in a pickle file (i.e., in the *cloud layer*), where the features and corresponding labels of the hand gesture sequences are organized. The features are then organized into sequences of frames, with each frame representing a window of historical data of the hand gesture sequence. We utilize the LSTM architecture, a type of recurrent neural network (RNN) designed to effectively model long-term dependencies in sequential

data. During training, the LSTM sub-model learns to capture intricate patterns and relationships within the hand gesture sequences. We employ techniques such as dropout regularization to prevent overfitting and optimize the model's generalization ability.

- **Sign Language Recognition:** This module is responsible for predicting the hand gestures based on the training information from both the CNN and LSTM models (i.e., *hybrid model*). Further details on the hybrid model are elaborated in Section 3.2.

3.2. Hybrid Model

Our proposed architecture is designed to gather the ArSL data that require translation using images or videos collected from cameras, displaying the corresponding text translation on screen in real time. We propose a hybrid model that consists of a CNN classifier to extract spatial features from ArSL data and an LSTM classifier to extract spatial and temporal characteristics to handle sequential data. To enable the user to use both images and videos, we created a function called "Sign Language Recognition Decision" to give our system the flexibility to use either images or videos. This function essentially chooses between the image processing sub-model (CNN) and the video processing sub-model (LSTM) depending on the input type. By integrating the strengths of CNNs and LSTMs, the system can leverage both spatial and temporal information simultaneously. The CNN extracts relevant visual features, and the LSTM processes these features in a sequential manner, capturing the dynamic aspects of sign language communication. This allows users to communicate naturally using gestures. More details on CNN and LSTM models are elaborated in the following subsection.

3.2.1. Convolutional Neural Network (CNN)

The CNN sub-model of our *hybrid model* is designed to detect human hands by leveraging the Google MediaPipe framework to identify the hand's 21 3D landmarks. These landmarks are crucial for understanding hand gestures and movements. For the proposed CNN sub-model, we utilized LeNet-5, a well-known CNN architecture frequently employed for image recognition tasks. This architecture comprises several layers: it begins with two convolutional layers, each followed by max-pooling layers, and concludes with three fully connected (linear) layers. As shown in Figure 2, the input to LeNet-5 is a grayscale image with dimensions of 32 × 32 × 1, where the "1" denotes a single channel for grayscale intensity. The first convolutional layer applies six filters of size 5 × 5 to the input image, producing an output shape of [−1, 6, 28, 28]. The subsequent max-pooling layers halve the spatial dimensions, resulting in feature maps of sizes [−1, 6, 14, 14] and [−1, 16, 5, 5], respectively. The final three linear layers have output shapes of [−1, 120], [−1, 84], and [−1, 10], with the last layer representing the output classes. LeNet-5 consists of 61,706 parameters in total, including weights and biases, distributed across the convolutional and linear layers.

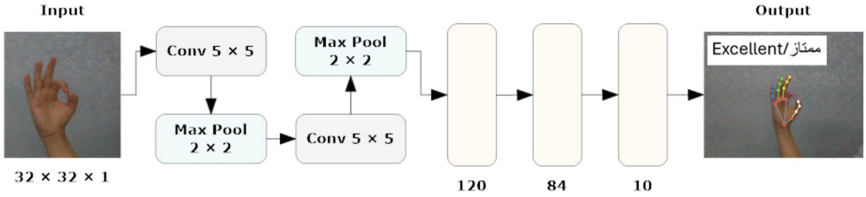


Figure 2. CNN model structure.

To train the CNN sub-model, we utilized pre-processed data stored in a pickle file. We then employed the Random Forest Classifier (RFC) to predict the output of these landmarks. Bootstrapping was used to sample a random subset of data and features used to train each tree in the forest. An aggregation of the predictions produced by every single decision tree results in the final RFC predictions. The final RFC prediction for hand gesture γ taking on sign language word class σ in the tree is explained in Equation (1):

$$\mathbb{P}_\tau[\sigma|\gamma,\Omega,\pi] = \sum_{\lambda\in\Lambda} \pi\lambda_\sigma \mu_\lambda(\gamma,\Omega), \tag{1}$$

where Ω denotes the parameter of the decision function δ, π denotes the class label distribution of all leaf nodes, Λ denotes a set of leaf nodes, $\pi\lambda_\sigma\mu_\lambda$ denotes the probability that the hand gesture belongs to a sign language word class σ given by leaf node λ, and (γ,Ω) denotes the probability of routing the hand gesture until leaf node λ. Ultimately, we can interpret this probability value as a weighted sum of the class distribution if we treat the possibility of reaching the leaf node as a weight. The decision function δ for split node ν is defined in Equations (2) and (3):

$$\delta_\nu(\gamma;\Omega) = \beta(f_\nu(\gamma;\Omega)), \tag{2}$$

$$\beta(\gamma) = \left(1+e^{-\gamma}\right)^{-1}, \qquad \textit{where} \qquad f_\nu(.;\Omega) : \Gamma \to \mathbb{R} \tag{3}$$

where $\beta(\gamma)$ denotes the sigmoid function, whose output can be used to calculate the possibility of moving to the left or right sub-tree in the RFC, and $f_\nu(.;\Omega)$ denotes the real-valued function parametrized by Ω.

The model's ability to learn hand landmarks is encoded by the parameter Ω. The parameters of a deep CNN, which are used to automatically train an appropriate hand landmark representation from incoming images, are represented by Ω in this paper. Every function f_ν can be thought of as a deep network's linear output unit. The final prediction of the CNN model, delivered by the forest $F = \{\tau_1, \tau_2, \ldots, \tau_\gamma\}$ for the hand landmark γ, is calculated as shown in Equation (4):

$$\mathbb{P}_F[\sigma|\gamma] = \frac{1}{x}\sum_{n=1}^{X} \mathbb{P}_{\tau_n}[\sigma|\gamma], \tag{4}$$

where γ denotes the hand gesture, σ denotes the sign language word class, x denotes the number of trees in the forest, and $\mathbb{P}_{\tau_n}$ represents the probability that the hand gesture belongs to sign language word class σ given by the n^{th} tree. The average of the prediction made by each tree in the forest eventually determines the final prediction.

3.2.2. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to effectively model long-term dependencies in sequential data [39–41]. For the proposed LSTM sub-model, we employ a stack structure, which involves arranging multiple LSTM layers in sequence, where the output of each LSTM layer serves as the input for the next. This deep architecture allows for capturing more intricate patterns in sequential data by utilizing both LSTM and dense layers. As illustrated in Figure 3, the first LSTM layer has 64 units, returns sequences, and uses the hyperbolic tangent function (*tanh*) for activation. The second LSTM layer includes 128 units, returns sequences, and also uses *tanh* for activation. The third LSTM layer, with 64 units, does not return sequences and performs *tanh* activation. Following the LSTM layers are the dense layers: the first dense layer has 64 units and uses *tanh* activation, the second dense layer has 32 units with *tanh* activation, and the final dense layer has units equal to the number of actions, employing the Softmax function to constrain the outputs between 0 and 1, ensuring their sum is always 1.

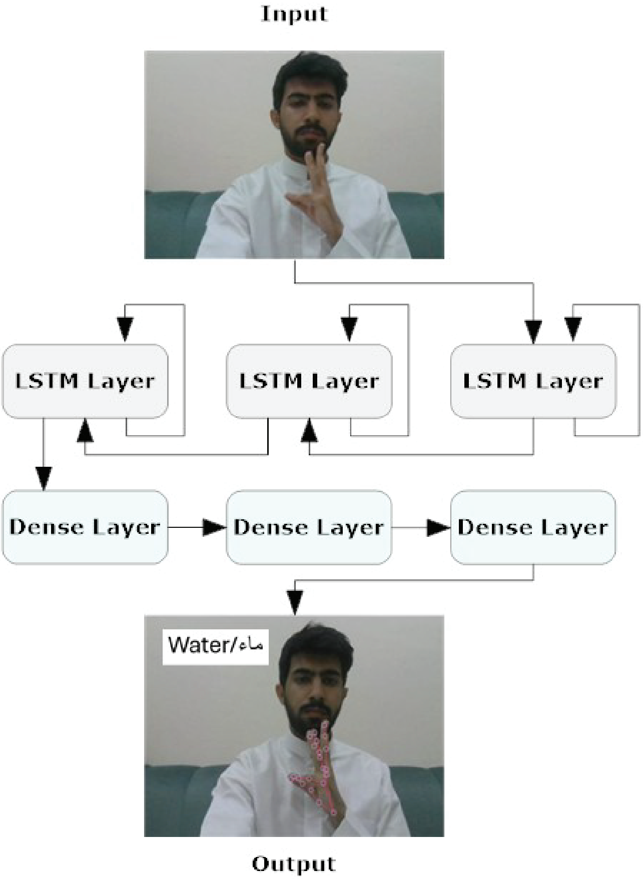


Figure 3. LSTM model structure.

During training, the LSTM sub-model learns to capture intricate patterns and relationships within the hand gesture sequences. In this paper, LSTM model is employed to learn hand gesture sequences and to recognize the correspondent sign language translation. leveraging the Google MediaPipe framework to extract input representations, the structure of LSTM is expressed in the following manner:

A form of RNN called LSTM architecture is intended to efficiently model long-term dependencies in sequential data [15,42,43]. During training, the LSTM sub-model picks up on the complex correlations and patterns found in the hand gesture sequences. To learn hand gesture sequences and recognize the corresponding sign language translation, this paper uses an LSTM model. Exploiting Google MediaPipe framework to extract input representations, the structure of LSTM consists of several gates including the input gate, the forget gate, the output gate, and the cell input vector. The input gate is described in Equation (5) as follows:

$$\alpha_t = \psi(\omega_\alpha \cdot [h_{t-1}, x_t] + b_\alpha), \tag{5}$$

where the input gate, denoted as α, is controlled by the time step, denoted as t. ψ denotes the logistic sigmoid function. The last hidden state, denoted as $h_[t-1]$, and the current input, denoted as x_t, are weighted using the matrix weight ω. b_α denotes the biases of the input gate. The forget gate is described in Equation (6), as follows:

$$\varphi_t = \psi(\omega_\varphi \cdot [h_{t-1}, x_t] + b_\varphi), \tag{6}$$

where the forget gate, denoted as φ, is controlled by the time step, denoted as t. Similar to the input gate, ψ denotes the logistic sigmoid function. The last hidden state, denoted as h_{t-1}, and the current input, denoted as x_t, are weighted using the matrix weight ω. b_φ denotes the biases of the forget gate. As we can observe, the input gate's mechanism is identical to this one, but it uses a completely different set of weights. The output gate is expressed in Equation (7), as follows:

$$\beta_t = \psi(\omega_\beta \cdot [h_{t-1}, x_t] + b_\beta), \tag{7}$$

where the output gate, denoted as β, is also controlled by the time step, denoted as t. Similar to the input and forget gates, ψ denotes the logistic sigmoid function. The last hidden state, denoted as h_{t-1}, and the current input, denoted as x_t, are weighted using the matrix weight ω. b_φ denotes the biases of the output gate. The cell input vector is shown in Equations (8) and (9), as follows:

$$\varepsilon_t = \varphi_t \odot \varepsilon_{t-1} + t_i \odot tanh(\omega_\varepsilon \cdot [h_{t-1}, x_t] + b_\varepsilon), \tag{8}$$

$$h_t = \beta_t \odot tanh(\varepsilon_t), \tag{9}$$

where *tanh* denotes the tangent function used to transform the data into a normalized encoding of the data. $\odot$ represents the element-wise product of two vectors.

4. Implementation

For the implementation, we utilized Google Cloud Computing Services (Google Cloud) (https://cloud.google.com/, accessed on 28 February 2024). Within this environment, we deployed virtual machines running the Ubuntu Operating System version 20.04, sourced from the Ubuntu OS Cloud Marketplace (https://console.cloud.google.com/marketplace/product/ubuntu-os-cloud/ubuntu-focal, accessed on 28 February 2024). These virtual machines were configured with the Docker platform (https://cloud.google.com/compute/docs/containers, accessed on 28 February 2024) to execute the application container. In particular, we employed the standard `c3d-Standard-4` configuration on Google Cloud, with each virtual machine featuring 4 VCPUs and 16 gigabytes (GB) of memory. Our system architecture consisted of three such virtual machines, each assigned distinct responsibilities for each layer in our architecture, including the data acquisition, sign language recognition, and cloud layers, as illustrated in Figure 1. Leveraging Google Cloud facilitated the setup and management of our system architecture, providing reliability, speed, and scalability. Additionally, Google Cloud offers a range of services and features tailored to our system architecture requirements and use cases. We also leveraged Google Cloud's security and governance services to safeguard our data and applications against unauthorized access and potential threats. The configuration parameters and corresponding values chosen for setting up and managing our system architecture on Google Cloud are detailed in Table 1.

Table 1. List of configuration parameters and values.

Parameter	Value
Cloud service provider	Google Cloud
Instance type	Ubuntu 20.04
Operating system	CentOS Linux
CPU	Intel Xeon Platinum 8481C Processor
RAM	16 GB
Disk size	32 TB

Upon initiating the `c3d-Standard-4` instance, we proceeded to install Docker version 4.28.0 on the virtual machine (https://docs.docker.com/desktop/release-notes/#4280, accessed on 29 February 2024). Subsequently, we established three containers utilizing Arch Linux via Docker. Each container served a distinct purpose within our system architecture: the first container managed the data acquisition layer, the second container handled the sign language recognition layer, and the third container oversaw the cloud layer functionality. The data acquisition layer was responsible for gathering images or videos from various sources such as webcams, smartphone cameras, and wearable gadget cameras. The sign language recognition layer processed these data to interpret hand gestures. These containers were equipped with a Python interpreter version 3.12.2 (https://www.python.org/downloads/release/python-3122/, accessed on 1 March 2024) and NodeJS version 20.11.1 (https://nodejs.org/en/blog/release/v20.11.1, accessed on 1 March 2024), enabling us to develop and execute code for both the data acquisition and sign language recognition layers. Python facilitated the creation of concise and comprehensible code capable of handling intricate tasks and data structures. Meanwhile, NodeJS permitted the utilization of JavaScript for both front-end and back-end development, streamlining our codebase and enhancing performance. The final container utilized a Docker Compose file comprising an image for MySQL to effectively manage and store the data (https://hub.docker.com/_/mysql, accessed on 29 February 2024).

4.1. Dataset Description

In this paper, we employed a hybrid model that leverages deep learning techniques to achieve high performance and accuracy in gesture recognition tasks. To accomplish this, we utilized a large-scale dataset consisting of images and videos captured using built-in webcams with a resolution of 480p. For the CNN model, we collected 400 images for each of the 10 targeted words. Each image has dimensions of 640 × 480 pixels, ensuring a detailed representation of hand gestures. Additionally, for the LSTM model, we obtained 50 videos for each word (i.e., 10 words), totaling 500 videos. These videos, recorded with the same 480p webcam, provide dynamic visual sequences for temporal analysis. We used the most universal and commonly used words for sign language, like hello, water, teacher, work, etc. The translations of these words were taken from the Handspeak website (https://www.handspeak.com/, accessed on 12 February 2024) and the Saudi Sign Language website (https://sshi.sa/, accessed on 12 February 2024). The Handspeak website is an online resource for sign language, and the Saudi Sign Language website is a platform affiliated with the Saudi Society for Hearing Impairment. The diverse nature and scale of our dataset enabled robust training and evaluation of the hybrid model for accurate gesture recognition tasks.

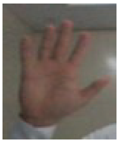
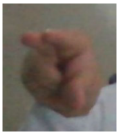
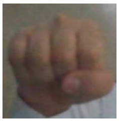
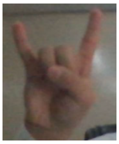
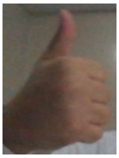
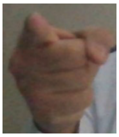
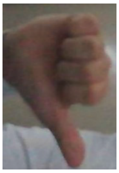
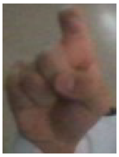
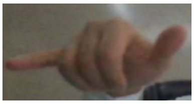
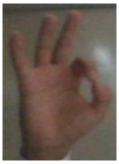
4.2. Data Labeling and Model Training

We divided the dataset into 80% for training and 20% for testing to validate the proposed hybrid model.

4.2.1. CNN Sub-Model

The dataset for training the CNN sub-model comprised images we captured, each depicting a specific hand gesture. We utilized MediaPipe to extract hand features from these images, focusing on 21 hand landmarks. These features were saved as `.npy` files in directories corresponding to each of the 10 distinct gesture labels in our dataset. Each label category contained 400 images, resulting in a balanced dataset of 4000 images. Of these, 3200 hand landmark images were manually labeled and categorized based on the selected 10 words (as shown in Table 2) to train the CNN sub-model of the proposed hybrid model. The remaining 800 hand landmark images were used to evaluate the CNN sub-model.

Table 2. CNN data labeling.

Label	Name	Gesture
0	Hello/مرحبا	
1	No/لا	
2	Agree/أتفق	
3	Awesome/مذهل	
4	Good/جيد	
5	You/أنت	
6	Bad/سيئ	
7	Question/سؤال	
8	Not sure/لست متأكد	
9	Excellent/ممتاز	

As detailed in Table 3, the CNN sub-model was trained using the following hyperparameters: a learning rate of 0.001 to ensure stable convergence, a batch size of 32 to balance memory usage and training speed, and a total of 50 epochs to allow for thorough training and convergence. We used the Adam optimizer, known for its efficiency and adaptive learning rates, and employed cross-entropy loss as the loss function, which is suitable for our classification task. To prevent overfitting and enhance the model's generalization ability, we incorporated a dropout rate of 0.5. Dropout is a regularization technique that involves randomly ignoring selected neurons during training, which helps the model avoid relying too heavily on specific features or neurons and promotes robustness.

Table 3. CNN sub-model hyperparameter and loss function settings.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	32
Number of Epochs	50
Optimizer	Adam
Loss function	Cross-Entropy
Dropout Rate	0.5
Data Augmentation	No

4.2.2. LSTM Sub-Model

The dataset comprised recorded sequences of actions stored in directories corresponding to each specific action. It included 10 actions, each represented by 50 videos. Each sequence consisted of 30 frames, with each frame represented by a 1662-dimensional feature vector. These files were loaded into memory, and the sequences and their labels were compiled into arrays. We captured the videos in this dataset; 400 of the collected hand gesture sequence videos were manually labeled and categorized according to 10 selected words (as shown in Table 4) to train the LSTM sub-model of the proposed hybrid model, which was trained for 50 epochs. Additionally, 100 hand gesture sequence videos were used to test the LSTM sub-model.

Table 5 provides details of the LSTM sub-model's training, including the parameters for the LSTM layers and dense layers. The first LSTM layer outputs a shape of (None, 30, 64) with 442,112 parameters, followed by a second LSTM layer with an output shape of (None, 30, 128) and 98,816 parameters. The third LSTM layer produces an output shape of (None, 64) with 49,408 parameters. This is followed by three dense layers: the first dense layer has an output shape of (None, 64) with 4,160 parameters, the second dense layer has an output shape of (None, 32) with 2,080 parameters, and the third dense layer has an output shape of (None, 10) with 330 parameters. In total, the model has 596,906 parameters.

Table 4. LSTM data labeling.

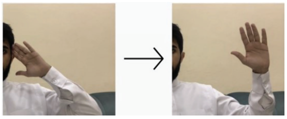
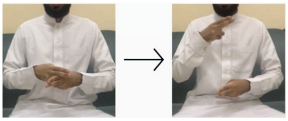
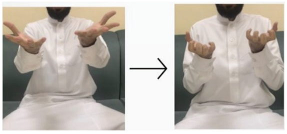
Label	Name	Gesture
0	Peace be upon you/ السلام عليكم	
1	My name/ أسمي	
2	I want/ أريد	

Table 4. *Cont.*

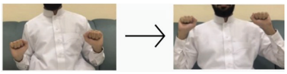
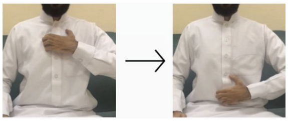
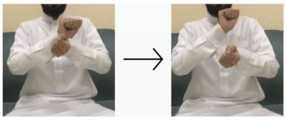
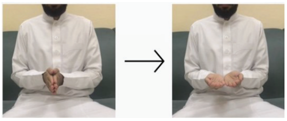
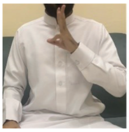
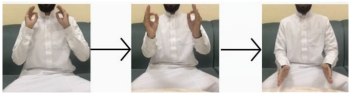
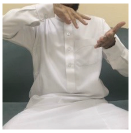
Label	Name	Gesture
3	An exercise/تمرين	
4	Hungry/جائع	
5	Work/عمل	
6	Book/كتاب	
7	Water/ماء	
8	Teacher/معلم	
9	Teaches/يدرس	

Table 5. LSTM sub-model's layers and parameter setup.

Layer (Type)	Output Shape	Param. #
Lstm 1 (LSTM)	(None, 30, 64)	442,112
Lstm 2 (LSTM)	(None, 30, 128)	98,816
Lstm 3 (LSTM)	(None, 64)	49,408
Dense 1(Dense)	(None, 64)	4160
Dense 2 (Dense)	(None, 32)	2080
Dense 3 (Dense)	(None, 10)	330
Total params:		596,906

5. Experimental Results

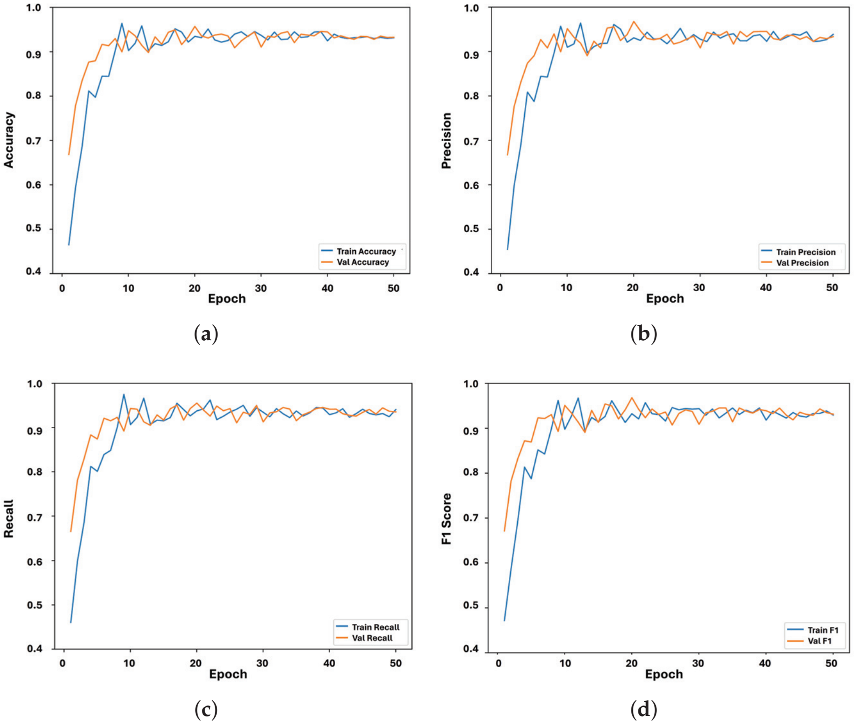
To demonstrate our proposed hybrid model's applicability, we conducted several experiments to evaluate the performance of each sub-model of our proposed hybrid model, including the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models.

5.1. Evaluation Metrics

In this section, we outline the evaluation criteria used to gauge the effectiveness of the hybrid model, which includes the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. Specifically, we examine important metrics such as accuracy, precision, recall, F1 score, confusion matrix, and loss. These metrics are essential for gaining a comprehensive understanding of the hybrid model's predictive abilities and the sub-models' ability to generalize. Alongside the traditional performance metrics, assessing loss provides insights into the optimization process and the convergence of the sub-models. It is worth emphasizing that all metrics were assessed using a distinct validation dataset, separate from the training data, to ensure the robustness and reliability of the hybrid model performance assessment.

5.2. CNN Evaluation Metrics

Figure 4 illustrates the results of the performance of the CNN sub-model on the 10 words listed in Table 2. For this experiment, we trained the CNN sub-model using 3200 images (i.e., for hand landmark images) and validated it using 800 images. The accuracy of the CNN sub-model was 94.40% (see Figure 4a). The reason for this is the adequate amount of data for training, where the model could understand the hand gestures using the hand landmarks. The main purpose for setting the number of epochs to 50 was to leave room for the CNN sub-model to understand the targeted hand gestures. Additionally, the CNN sub-model achieved a precision of 95.00% (see Figure 4b) and recall and F1 scores of 94.40% and 94.90%, respectively, as shown in Figure 4c,d. The accuracy, precision, recall, and F1 scores consistently fell within the range of 91.00% to 95.00%. This uniformity in performance across these metrics indicates balanced model behavior. The similarity in accuracy, precision, recall, and F1 score can be attributed to the balanced nature of the dataset, where each class is adequately represented, facilitating equitable performance evaluation across all classes.



(**a**) (**b**) (**c**) (**d**)

Figure 4. CNN sub-model performance evaluation. (**a**) CNN sub-model accuracy; (**b**) CNN sub-model precision; (**c**) CNN sub-model recall; (**d**) CNN sub-model F1 score.

Figure 5 presents the confusion matrix derived from the validation results of the CNN sub-model. This matrix offers a comprehensive visual representation of the CNN sub-model's classification performance, highlighting the distribution of true positive, true negative, false positive, and false negative predictions across 10 different classes (i.e., the 10 labels for the 10 words listed in Table 2). Examining the confusion matrix provides crucial insights into the CNN sub-model's ability to accurately classify instances within each class. As we can see, the true positives fell within the range of 82 to 97 when the CNN sub-model predicted the targeted word correctly. We can make an interesting observation from the confusion matrix: the CNN sub-model scored 89 and 92 TPs for labels 3 and 8, respectively. This is an indication that the CNN sub-model's recognition performance was affected by viewing different angles. Furthermore, Figure 6 shows the loss evaluation for the training and validation of the CNN sub-model. The observed discrepancy in loss between the training and validation phases indicates that during the validation phase, the CNN sub-model became increasingly confident in interpreting hand gestures using hand landmarks as the number of epochs progressed.

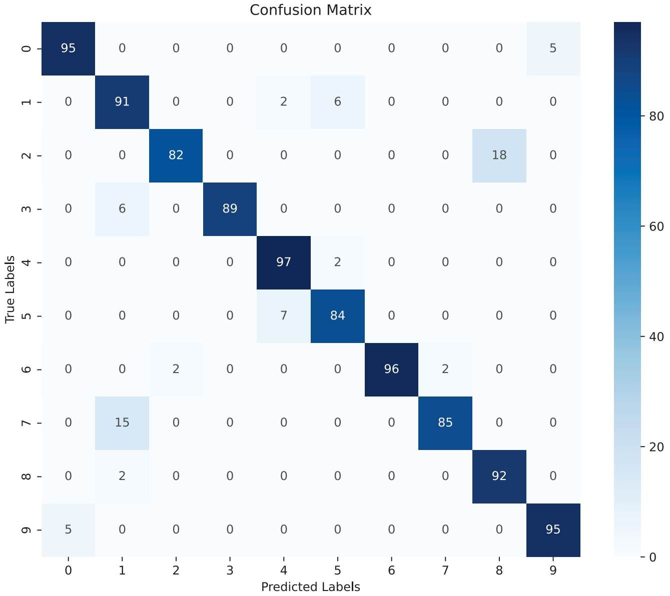


Figure 5. CNN sub-model confusion matrix.

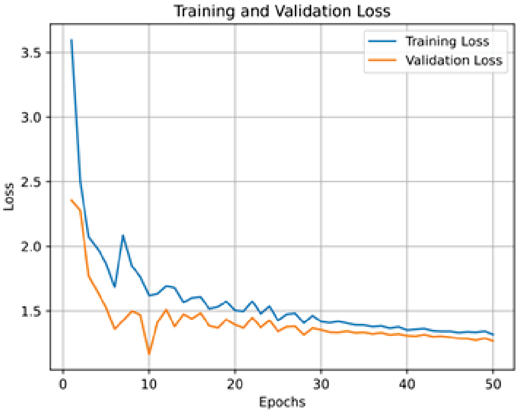
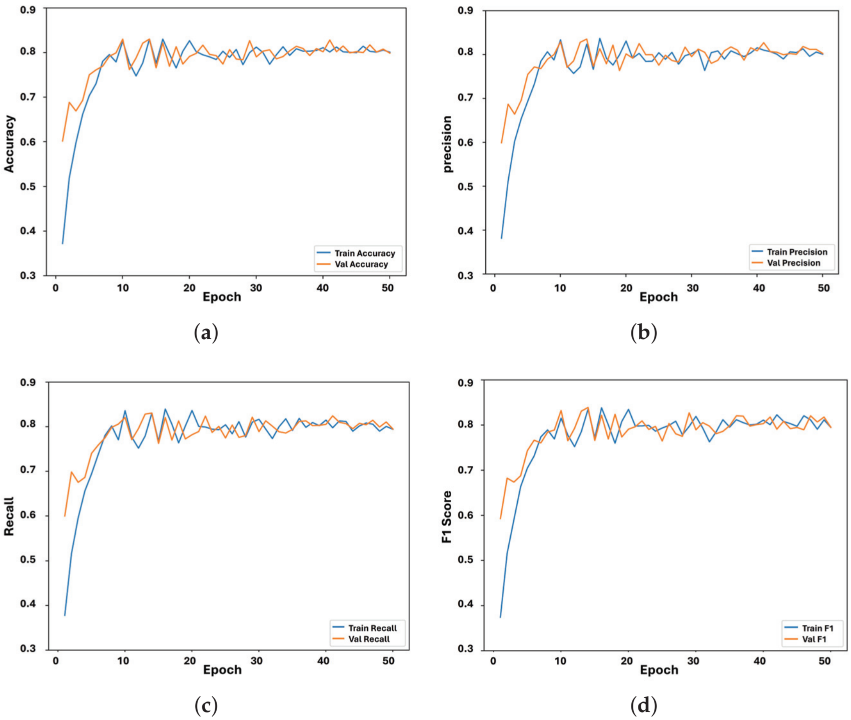


Figure 6. CNN sub-model validation loss.

5.3. LSTM Evaluation Metrics

Figure 7 illustrates the results of the performance of the LSTM sub-model on the 10 words listed in Table 4. For this experiment, we trained the LSTM sub-model using 400 videos (i.e., for hand gesture sequences) and validated it using 100 videos. As we can see, the accuracy of the LSTM sub-model was 82.70% (see Figure 7a). The reason for this is the adequate amount of data for training, where the model could understand the hand gesture sequences. For this experiment, we also used 50 epochs to allow the LSTM sub-model to understand the targeted hand gesture sequences. We can also observe that the LSTM sub-model reached a precision of 84.20% (see Figure 7b) and recall and F1 scores of 82.50% and 82.70%, respectively, as shown in Figure 7c,d. The accuracy, precision, recall, and F1 scores consistently fell within the range of 78.50% to 82.70%. The consistency in performance across these metrics suggests well-balanced model behavior. The similarity in accuracy, precision, recall, and F1 score can be attributed to the balanced composition of the dataset, ensuring sufficient representation of each class. This balanced representation enabled fair evaluation of performance across all classes.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. LSTM sub-model performance evaluation. (**a**) LSTM sub-model accuracy; (**b**) LSTM sub-model precision; (**c**) LSTM sub-model recall; (**d**) LSTM sub-model F1 score.

Figure 8 presents the confusion matrix derived from the validation results of the LSTM sub-model. This matrix offers a comprehensive visual representation of the LSTM sub-model's classification performance, highlighting the distribution of true positive, true negative, false positive, and false negative predictions across 10 different classes (i.e., the 10 labels for the 10 words listed in Table 4). Examining the confusion matrix provides crucial insights into the LSTM sub-model's ability to accurately classify instances within each class. As we can see, the true positives fell within the range of 56 to 95 when the LSTM sub-model correctly predicted the targeted word. Furthermore, Figure 9 shows the loss evaluation for the training and validation of the LSTM sub-model. We can see fluctuations in the loss outcomes between the training and validation phases, which can be attributed to the LSTM sub-model's focus on extracting temporal features, such as motion signs, rather than spatial features, such as static signs, as observed in the CNN sub-model. Additionally, the loss results in the validation phase of the CNN sub-model (see Figure 6) exhibit superior performance compared to those of the LSTM sub-model, primarily due to this distinction in feature extraction.

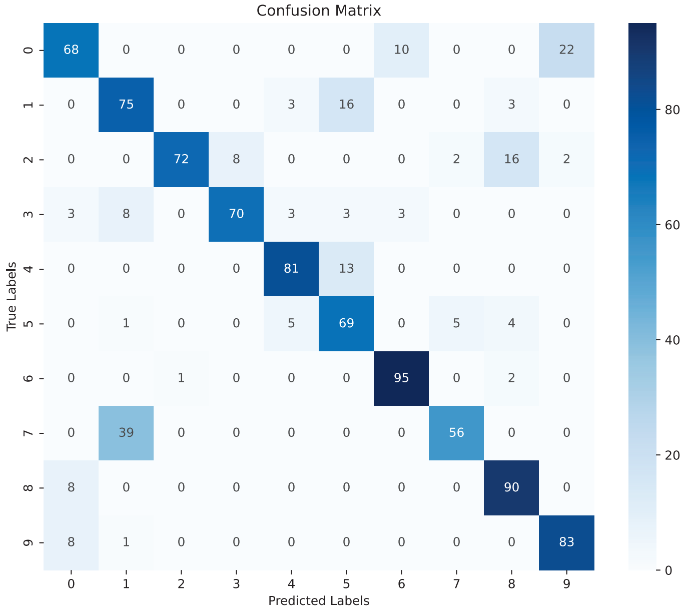


Figure 8. LSTM sub-model confusion matrix.

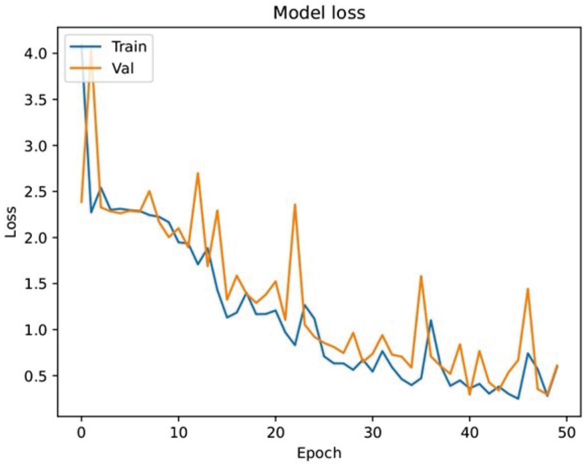


Figure 9. LSTM sub-model validation loss.

6. Conclusions and Future Work

We examined the issue of the pronounced deficiency of sign language interpreters in the Kingdom of Saudi Arabia, highlighting the notable gap in the ratio of interpreters to individuals with hearing impairments compared to other regions like California, USA. This study revealed that this scarcity presents a significant obstacle in delivering services to the deaf community in public spaces. In this paper, we propose a hybrid model designed to capture both spatial and temporal elements of sign language. This hybrid model comprises a CNN classifier to extract spatial features from sign language data and an LSTM classifier to capture both the spatial and temporal characteristics essential for sequential data processing. By automating ArSL translation in real time between sign language and spoken or written language, this hybrid model aims to address the interpreter shortage.

To demonstrate the viability of our proposed hybrid model, we created a dataset of 20 different words, comprising 4000 images for 10 static gesture words and 500 videos for 10 dynamic gesture words. Our hybrid model showcased promising performance, with the CNN and LSTM classifiers achieving accuracy rates of 94.40% and 82.70%, respectively. One implication of our research is the superior performance of the CNN compared to LSTM, attributed to LSTM's emphasis on extracting temporal features. At the same time, the CNN focuses on spatial features, such as static signs. In future work, we aim to expand the number of words in the datasets for both models in terms of both images and videos and use different viewing angles to enhance the performance of the sub-models utilized in our hybrid model. This research work holds the potential to enhance the availability of translation services, catering to the needs of the deaf community, thereby fostering effective communication, enhancing accessibility, and promoting solidarity with this significant segment of society.

Author Contributions: Conceptualization, T.H.N., A.N., A.F.A. and A.F.; methodology, T.H.N. and A.N.; software, A.F.A., A.F., R.A. and A.S.A.; validation, T.H.N., A.F.A. and A.F.; formal analysis, T.H.N., A.N., R.A. and A.S.A.; investigation, T.H.N., G.A. and A.A.; resources, T.A. and A.A.; data curation, A.F.A., A.F., R.A. and A.S.A.; writing—original draft preparation, T.H.N. and A.N.; writing—review and editing, G.A. and T.A.; visualization, T.H.N. and A.N.; supervision, T.H.N. and A.N.; project administration, T.H.N.; funding acquisition, A.N., G.A. and T.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 data is available on the following link (https://github.com/AhmedIbrahim110/Dataset).

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

References

1. Rastgoo, R.; Kiani, K.; Escalera, S. Sign language recognition: A deep survey. *Expert Syst. Appl.* **2021**, *164*, 113794. [CrossRef]
2. Costello, E. *Random House Webster's Compact American Sign Language Dictionary*; Penguin Random House: New York, NY, USA, 2008.
3. Kumar, P.; Gauba, H.; Roy, P.P.; Dogra, D.P. A multimodal framework for sensor based sign language recognition. *Neurocomputing* **2017**, *259*, 21–38. [CrossRef]
4. Hassan, M.; Assaleh, K.; Shanableh, T. Multiple proposals for continuous arabic sign language recognition. *Sens. Imaging* **2019**, *20*, 4. [CrossRef]
5. Ministry of Health, S.A. The Deaf and Sign Language. Available online: https://www.moh.gov.sa/en/Ministry/Information-and-services/Pages/Sign-language.aspx (accessed on 15 February 2024).
6. Ministry of Health, S.A. We Are with You. Available online: https://www.moh.gov.sa/en/Ministry/Projects/with-you/Pages/default.aspx (accessed on 10 February 2024).
7. Center for Strategic and International Studies (CSIS). Reading the Signs: Diverse Arabic Sign Languages. Available online: https://www.csis.org/analysis/reading-signs-diverse-arabic-sign-languages (accessed on 5 January 2024).
8. Alzohairi, R.; Alghonaim, R.; Alshehri, W.; Aloqeely, S. Image based Arabic sign language recognition system. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 185–194. [CrossRef]
9. Zakariah, M.; Alotaibi, Y.A.; Koundal, D.; Guo, Y.; Elahi, M.M. Sign language recognition for Arabic alphabets using transfer learning technique. *Comput. Intell. Neurosci.* **2022**, *2022*, 4567989. [CrossRef]

10. Wadhawan, A.; Kumar, P. Sign language recognition systems: A decade systematic literature review. *Arch. Comput. Methods Eng.* **2021**, *28*, 785–813. [CrossRef]
11. Noor, T.H.; Noor, A.; Elmezain, M. Poisonous Plants Species Prediction Using a Convolutional Neural Network and Support Vector Machine Hybrid Model. *Electronics* **2022**, *11*, 3690. [CrossRef]
12. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote. Sens.* **2021**, *173*, 24–49. [CrossRef]
13. Wang, B.; Chen, Y.; Yan, Z.; Liu, W. Integrating Remote Sensing Data and CNN-LSTM-Attention Techniques for Improved Forest Stock Volume Estimation: A Comprehensive Analysis of Baishanzu Forest Park, China. *Remote Sens.* **2024**, *16*, 324. [CrossRef]
14. Almukhalfi, H.; Noor, A.; Noor, T.H. Traffic management approaches using machine learning and deep learning techniques: A survey. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108147. [CrossRef]
15. Noor, T.H.; Almars, A.M.; El-Sayed, A.; Noor, A. Deep learning model for predicting consumers' interests of IoT recommendation system. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 161–170. [CrossRef]
16. Abib, G.; Castel, F.; Satouri, N.; Afifi, H.; Said, A.M. Survey and Enhancements on Deploying LSTM Recurrent Neural Networks on Embedded Systems. In Proceedings of the ICC 2023-IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023; pp. 949–953.
17. Koller, O. Quantitative survey of the state of the art in sign language recognition. *arXiv* **2020**, arXiv:2008.09918.
18. Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 131–153. [CrossRef]
19. Mohandes, M.; Deriche, M.; Liu, J. Image-based and sensor-based approaches to Arabic sign language recognition. *IEEE Trans. Hum. Mach. Syst.* **2014**, *44*, 551–557. [CrossRef]
20. Aiouez, S.; Hamitouche, A.; Belmadoui, M.S.; Belattar, K.; Souami, F. Real-time Arabic Sign Language Recognition based on YOLOv5. In Proceedings of the IMPROVE, Online Streaming, 22–24 April 2022; pp. 17–25.
21. Alawwad, R.A.; Bchir, O.; Ismail, M.M.B. Arabic sign language recognition using Faster R-CNN. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 692–700. [CrossRef]
22. Elhagry, A.; Elrayes, R.G. Egyptian sign language recognition using cnn and lstm. *arXiv* **2021**, arXiv:2107.13647.
23. Hdioud, B.; Tirari, M.E.H. A Deep Learning based Approach for Recognition of Arabic Sign Language Letters. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 424–429. [CrossRef]
24. Rivera-Acosta, M.; Ruiz-Varela, J.M.; Ortega-Cisneros, S.; Rivera, J.; Parra-Michel, R.; Mejia-Alvarez, P. Spelling correction real-time American sign language alphabet translation system based on YOLO network and LSTM. *Electronics* **2021**, *10*, 1035. [CrossRef]
25. Dutta, K.K.; Bellary, S.A.S. Machine learning techniques for Indian sign language recognition. In Proceedings of the 2017 International Conference on Current Trends in Computer, Electrical, Electronics, and Communication (CTCEEC), Mysore, India, 8–9 September 2017; pp. 333–336.
26. Sako, S.; Hatano, M.; Kitamura, T. Real-time Japanese sign language recognition based on three phonological elements of sign. In Proceedings of the HCI International 2016–Posters' Extended Abstracts: 18th International Conference, HCI International 2016, Toronto, ON, Canada, 17–22 July 2016; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2016; pp. 130–136.
27. Uyyala, P. Sign Language Recognition Using Convolutional Neural Networks. *J. Interdiscip. Cycle Res.* **2022**, *14*, 1198–1207.
28. Vyavahare, P.; Dhawale, S.; Takale, P.; Koli, V.; Kanawade, B.; Khonde, S. Detection and interpretation of Indian Sign Language using LSTM networks. *J. Intell Syst. Control* **2023**, *2*, 132–142. [CrossRef]
29. Shurid, S.A.; Amin, K.H.; Mirbahar, M.S.; Karmaker, D.; Mahtab, M.T.; Khan, F.T.; Alam, M.G.R.; Alam, M.A. Bangla sign language recognition and sentence building using deep learning. In Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 16–18 December 2020; pp. 1–9.
30. Daniels, S.; Suciati, N.; Fathichah, C. Indonesian sign language recognition using YOLO method. In Proceedings of the IOP Conf. on Materials Science and Engineering, Yogyakarta, Indonesia, 13–14 November 2020; pp. 1–9.
31. Ko, S.K.; Son, J.G.; Jung, H. Sign language recognition with recurrent neural network using human keypoint detection. In Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems, Honolulu, HI, USA, 9–12 October 2018; pp. 326–328.
32. Deng, Z.; Leng, Y.; Chen, J.; Yu, X.; Zhang, Y.; Gao, Q. TMS-Net: A multi-feature multi-stream multi-level information sharing network for skeleton-based sign language recognition. *Neurocomputing* **2024**, *572*, 127194. [CrossRef]
33. Zuo, R.; Wei, F.; Mak, B. Natural language-assisted sign language recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 14890–14900.
34. Ryumin, D.; Ivanko, D.; Ryumina, E. Audio-visual speech and gesture recognition by sensors of mobile devices. *Sensors* **2023**, *23*, 2284. [CrossRef] [PubMed]
35. Qi, W.; Fan, H.; Xu, Y.; Su, H.; Aliverti, A. A 3d-CLDNN based multiple data fusion framework for finger gesture recognition in human-robot interaction. In Proceedings of the 2022 4th international conference on control and robotics (ICCR), Guangzhou, China, 22–24 October 2022; pp. 383–387.
36. Bora, J.; Dehingia, S.; Boruah, A.; Chetia, A.A.; Gogoi, D. Real-time assamese sign language recognition using mediapipe and deep learning. *Procedia Comput. Sci.* **2023**, *218*, 1384–1393. [CrossRef]

37. Eunice, J.; Sei, Y.; Hemanth, D.J. Sign2Pose: A Pose-Based Approach for Gloss Prediction Using a Transformer Model. *Sensors* **2023**, *23*, 2853. [CrossRef] [PubMed]
38. Sincan, O.M.; Keles, H.Y. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access* **2020**, *8*, 181340–181355. [CrossRef]
39. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
40. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
41. Alqurafi, A.; Alsanoosy, T. Measuring Customers' Satisfaction Using Sentiment Analysis: Model and Tool. *J. Comput. Sci.* **2024**, *20*, 419–430. [CrossRef]
42. Van Houdt, G.; Mosquera, C.; Nápoles, G. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [CrossRef]
43. Bansal, M.; Goyal, A.; Choudhary, A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decis. Anal. J.* **2022**, *3*, 100071. [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.

MDPI

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-2814-2