

Article

Video Content Plagiarism Detection Using Region-Based Feature Learning

Xun Jin ^{1,†}, Su Yan ^{1,†}, Rongchun Chen ¹, Xuanyou Li ², De Li ^{1,*} and Yanwei Wang ^{3,*}¹ Department of Computer Science and Technology, Yanbian University, Yanji 133000, China; xunjia@ybu.edu.cn (X.J.); yansu623@189.cn (S.Y.); rcchanyi@gmail.com (R.C.)² Information Engineering Major, College of Communication Engineering, Jilin University, Changchun 130012, China; hyunwulee@126.com³ College of Intelligent Manufacturing, Changchun Sci-Tech University, Changchun 130012, China

* Correspondence: leader1223@ybu.edu.cn (D.L.); wangyanwei@cstu.edu.cn (Y.W.); Tel.: +86-151-6822-2907 (D.L.)

† These authors contributed equally to this work.

Abstract

Due to the continuous increase in copyright infringement cases of video content, the economic losses of copyright owners continue to rise. To improve the efficiency of plagiarism detection in video content, in this paper, we propose region-based video feature learning. The first innovation of this paper lies in the combination of temporal positional encoding and attention mechanisms to extract global features for weakly supervised model training. Self- and cross-attention mechanisms are combined to enhance similar features within and between videos by incorporating position coding to capture timing relationships between video frames. Global classification description is embedded for capturing global spatiotemporal information and combined with a weak supervised loss for model training. The second innovation is the frame sequence similarity calculation, which is composed of Chamfer similarity, coordinate attention mechanism, and residual connection, to aggregate similarity scores between videos. Experimental results show that the proposed method can achieve the mAP of 0.907 on the short video dataset from Douyin. The proposed method outperforms frame-level and video-level features in achieving higher detection accuracy, and further contributes to the improvement of video content plagiarism detection performance.

Keywords: video similarity detection; feature extraction; video copyright protection; deep learning



Academic Editor: Aryya Gangopadhyay

Received: 12 August 2025

Revised: 2 October 2025

Accepted: 10 October 2025

Published: 13 October 2025

Citation: Jin, X.; Yan, S.; Chen, R.; Li, X.; Li, D.; Wang, Y. Video Content Plagiarism Detection Using Region-Based Feature Learning. *Electronics* **2025**, *14*, 4011. <https://doi.org/10.3390/electronics14204011>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the large number of users on short video platforms and the massive daily uploads of videos, conducting comprehensive reviews for each video has become extremely difficult. This has led to many instances of copyright infringement going unnoticed during the review process. As a result, research on video infringement detection has become particularly important.

To avoid detection by plagiarism detection systems, plagiarists maliciously interfere with the video content by modifying it. These changes potentially alter some attributes of the work, posing challenges in detection and proof. However, similarities still exist between the contents of the works. This provides feasibility for plagiarism detection.

Traditional copyright protection techniques, such as digital watermarking, encryption, keyword matching, and image recognition, are not fully applicable to short video content [1–7]. Although newer copyright verification technologies such as neural network-based feature recognition, video fingerprinting, and blockchain have to some extent solved the problem of watermark concealment, they still rely on content creators or users to discover and report infringement in terms of copyright protection. Therefore, further exploration and research are needed to improve the accuracy and robustness of detection performance, in order to develop more comprehensive, formal, and effective technologies and methods for making reasonable and objective judgments on the copyright of short video works.

This study starts from the perspective of video content analysis, using deep learning neural network models to extract key features to enhance recognition efficiency. By analyzing similarities between features, the proposed model compares the similarity between videos, thereby making copyright determinations. This provides objective grounds for identifying suspicious infringing works.

In this paper, we propose a method for extracting region-based features from videos and employ a linear Transformer encoder to enhance learning ability. It combines temporal positional encoding and attention mechanisms to extract global features for weakly supervised model training. Subsequently, it uses a Convolutional Neural Network (CNN) with coordinate attention mechanisms to aggregate feature similarity graphs. Features are extracted from various intermediate layers of CNN to form frame-level regional feature representations. Data augmentation is performed by randomly shuffling these representations, simulating different variations in image content positions, to enhance model generalization capability and performance.

A self-attention mechanism and a cross-attention mechanism are integrated into a linear Transformer encoder. This consolidation helps handle dependencies in long sequences. Positional encoding is employed to represent temporal information between video frames. The approach emphasizes internal and mutual spatio-temporal relationships among videos to enhance features. Global classification description vectors are embedded to characterize global spatio-temporal information. These vectors are combined with weakly supervised loss methods to supervise model training.

To match frame sequence similarity features, we employ the Chamfer similarity (CS) calculation method. Additionally, we utilize grouped convolution with coordinate attention mechanism and residual connections. The goal is to aggregate similarity scores between videos. Utilizing label similarity loss information and triplet learning, we conduct end-to-end training. This approach directly computes similarity values of input videos, evaluated on standard benchmark datasets. Finally, we design a copyright detection method to achieve plagiarism detection and validate its effectiveness through experiments.

The main contributions of this paper are summarized as follows: (1) Through data augmentation and the fusion of Linear Transformer encoder, cross- and self-attention mechanisms, the expression ability of video features has been enhanced, and the accuracy of extracting similar content features has been improved. (2) By utilizing global spatio-temporal information, the model training was effectively supervised and achieved detection performance improvement. (3) By using Chamfer similarity calculation and aggregating video similarity maps, more robust video similarity calculation results can be obtained.

2. Related Works

Video watermarking technology embeds secret information into the video content to substantiate copyright ownership. The commonly used watermarking techniques are transform domain-based methods, modifying coefficients in that domain to embed water-

marks and subsequently generating watermarked videos through inverse transformation. Deep learning-based watermarking techniques utilize CNNs such as StegaStamp model proposed in [1]. These techniques capture camera distortions and model original transformations like color changes, blurs, and perspective distortions. A two-stage separable blind watermarking based on deep learning framework was proposed in [2] for enhancing robustness against attacks. In [3], CNN-based watermarking was proposed by using adversarial training to replace designated attack training. A CNN-based end-to-end residual watermarking framework called RedMark was proposed in [4] to achieve real-time blind watermark extraction. In [5], a stable video zero-watermarking method leveraged CNN feature extraction capabilities. Three-dimensional CNN-based video watermarking was proposed in [6] to exploit spatial and temporal correlations in videos. It employed multi-scale embedding techniques across multiple spatio-temporal scales for end-to-end network training. In [7], a video tracing solution based on Ethereum smart contract was proposed to track the origin and history of even multiply replicated video content. However, these technologies cannot detect plagiarized contents with some changes.

Content-based video fingerprinting technology transforms digital video content into unique identifiers to achieve video content identification and recognition. With this technology, we can achieve an effective video copyright protection and management. This method covers the extraction of video feature fingerprints across multiple dimensions. These features include color space features, temporal features, spatial features, spatio-temporal features, and deep neural network features. The method additionally establishes a video fingerprint database. By combining nearest neighbor search and re-ranking strategy for feature matching, it calculates the similarity between fingerprints to generate detection results. A compact representation method for videos based on sparse coding and CNNs was proposed in [8]. This method utilized CNNs to extract deep features from video frames and transformed them into fixed-length vectors through sparse coding. In [9], a video fingerprinting method was proposed by combining CNN-based deep features with manually designed visual features. Video frame sequences are encoded through parallel and hierarchical processing. This process aims to extract binary video fingerprints. In [10], spatial similarity, temporal similarity and local alignment information are combined for detecting and locating partial video duplications. The similarity matrix of video frames can be transmitted into CNN for training. It calculates corner-cutting similarity scores between videos. This process aims to obtain temporal similarity of frame sequence matching.

With the advancement of deep neural network technology, deep features have demonstrated superior performance over traditional handcrafted features. This superiority is evident in various fields such as video feature extraction and similarity detection. This advancement effectively promotes research progress in content-based video retrieval and locating duplicated video segments, while also driving the development of technologies for short video copyright protection. Video features can be classified into frame-level and video-level categories: Frame-level features assess the similarity between frames or construct similarity graphs to evaluate the similarity score of the entire video. Video-level features aggregate multiple frame-level features into a single representation, such as global vectors, content-related segments, hash codes and bag-of-words, used for similarity measurement [11–14]. A self-supervised network that utilizes segment-level features and hash codes to achieve efficient performance was proposed in [13]. Metric learning was used to capture global feature similarity in [14]. In [15], a region-based content retrieval method was proposed using maximum activation of convolutions (RMAC). This method uses multiple regional features to represent video contents. In [16,17], a Chamfer similarity computation was combined with CNN structures to capture spatio-temporal similarities between videos. Subsequent work in this field involves knowledge distillation and self-

supervised learning to train network models. In [18], supervised contrastive learning and Transformer were proposed to aggregate contextual information. An object detection task was employed to locate duplicate video segments within video-to-video similarity graphs in [19]. It optimized evaluation metrics and used feature enhancement with flexible supervision methods for end-to-end model training. Techniques such as dynamic programming [20] and graph-based temporal network structures [21] have also been used to optimize the matching of similar video segments. However, further research and optimization are needed for robust feature extraction, model training optimization and similarity computation methods. These efforts aim to advance copyright protection and strengthen the protection of rights for original content creators.

3. The Proposed Video Feature Learning Method

3.1. Overview of the Proposed Framework

This section introduces the region-based feature similarity learning for video copyright detection. First, the module for extracting regional features of videos is discussed. The first module receives positive, negative, and anchor videos, and then obtains frame-level features through CNN, pooling, PCA and normalization. Next, linear Transformer encoder and temporal positional encoding are incorporated to better capture temporal information in video content. Then, a global spatiotemporal feature extraction is proposed based on self-attention and cross-attention. Finally, the design of similarity calculation and multiple loss functions is discussed. Figure 1 depicts the overall framework diagram of the model.

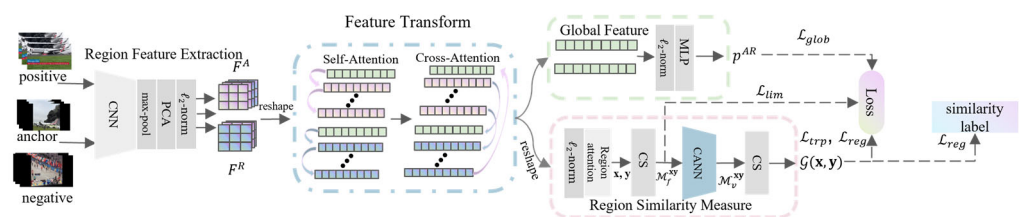


Figure 1. The flowchart for training model (pink represents FA, blue represents FR, and green represents global feature.).

First, the region feature extraction module plays a crucial role in extracting features from various regions of a video. These features capture local information such as texture and color. They are used to analyze video content at a fine-grained level. The feature transformation module utilizes mechanisms of self-attention and cross-attention. Self-attention dynamically assigns more weight to different regions to highlight important features. Cross-attention captures correlations between different regions, thereby improving the understanding of the overall structure and semantic information in a video. The global features of an entire video are extracted to provide higher-level semantic information. These features enable the model to effectively capture the overall content and style of the video during training. Additionally, these global features serve as supervision to guide network learning and optimization. In the region feature similarity calculation module, CS calculation is employed to assess the similarity between different regions in the video, aiding in precise comparison of video content and copyright detection. Finally, the loss functions play a role in supervision and optimization. By providing positive and negative sample pairs to the model, triplet loss and label supervision loss are introduced to constrain the model to learn more discriminative and distinguishable feature representations and enhance the generalization and robustness.

3.2. Region Feature Extraction

The quality of image features in video frames directly impacts the accuracy of similarity calculations. Typically, the feature maps extracted from the last layer are used to represent the entire image feature. However, images often contain multiple objects or scenes, with each region potentially having distinct features. This approach may lose some local fine-grained information, resulting in features that are less specific and discriminative. Additionally, the feature dimensions may become excessively large, containing redundant information.

Regional features capture details and characteristics of different areas within an image, and better distinguish between various objects, scenes, or styles. They include positional information which is important for tasks requiring context representation [22]. Integrating multi-scale features from different layers of a network model is crucial. Rather than relying solely on the final layer's feature map, this approach allows for the extraction of features from different sizes, and achieves better fine-grained detection results.

To fully model the multi-scale regional feature content, we adopt the RMAC across different layers. The approach involves applying max pooling operations with kernels of varying sizes and strides to different feature channels across different convolutional layers. This ensures that each convolutional layer obtains the same pooled result, specifically a 3×3 feature map size, to create regional features. The regional features are concatenated along the channel dimension to form a frame feature representation.

Due to concatenating all feature maps of the model, the resulting feature becomes high-dimensional. Neighboring dimensions may show minimal variation and contain a significant amount of redundant information. Therefore, PCA-whitening [23] is applied along each channel dimension. This method achieves a balance between preserving fine-tuned spatial information and achieving low feature dimensions.

Compared to encoding the entire image into a single feature representation, calculating similarity based on fine-grained feature vectors of region content can more accurately assess the similarity between similar contents. However, since the extracted regional features represent the content of different regions, in different positions of an image, there exists temporal sequence information. That is, the same features in different orders may represent different content of the image. A simple combination of the multi-head attention (MHA) mechanism results in different attention heads focusing only on fixed temporal positions of 3×3 features, neglecting weights for other positions. To address this, data augmentation is used during model training by randomly shuffling the order of region features within frame images, as shown in Figure 2. This approach allows each attention head a chance to focus on different region features. It improves the generalization ability, prevents overfitting, and better adapts to various content variations and noise.

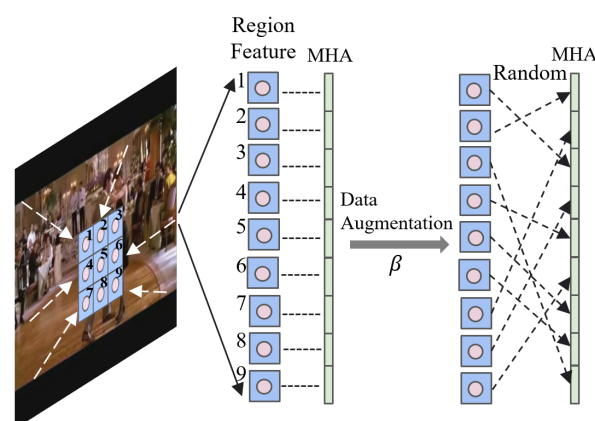


Figure 2. Schematic diagram of data augmentation.

In Figure 2, frame images undergo feature extraction to generate a region feature map, which is then arranged sequentially from left to right and top to bottom. β represents the probability controlling the feature data augmentation. During the training, if the value generated by a random probability seed is less than β , data augmentation is applied. This shuffles the order of region feature vectors in each frame image, mapping each feature vector to different attention heads for attention calculation. If the value is greater than or equal to β , the original data order is maintained. This approach effectively increases the diversity of training samples.

3.3. Linear Encoder Combined with Temporal Positional Encoding

Transformer encoder can be used to capture dependencies in video frame sequences. However, the computational complexity for n input video frames is $O(n^2)$. Additionally, training the model requires batching multiple videos, which increases computational and storage demands. Linear optimization can be used to reduce the model complexity to $O(n)$. In [24], a linear computation variant of the dot-product attention layer was used to optimize the Transformer's computational complexity from $O(n^2)$ to $O(n)$. This achieved good performance in visual tasks such as image feature localization, validating the effectiveness of linear optimization. Similarly, this paper adopts the linear optimization approach with linear Transformer's encoder structure. This structure models long-term temporal dependencies in extracted video region features. The structure of linear Transformer encoder and linear attention mechanism is shown in Figure 3.

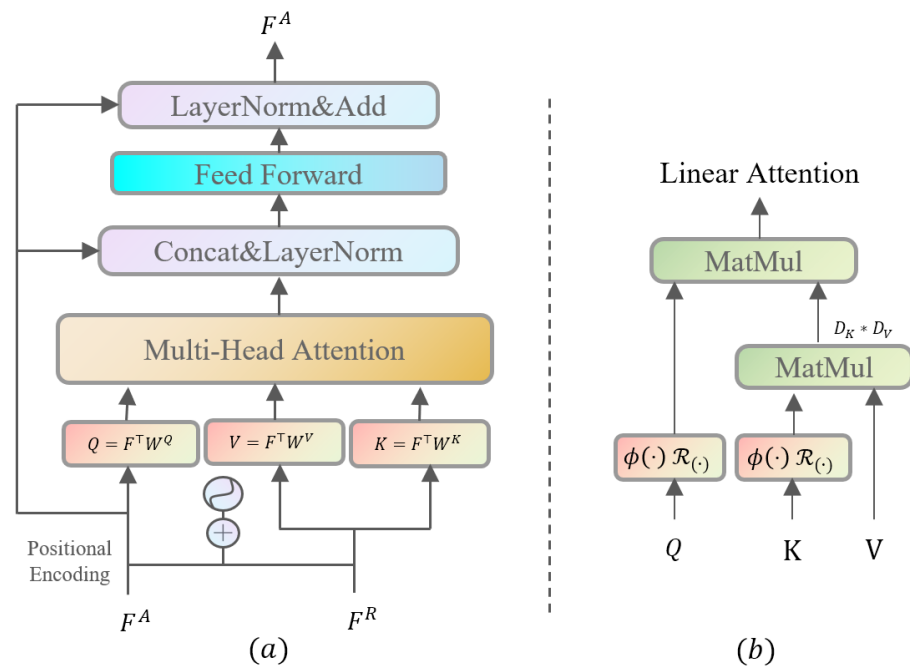


Figure 3. The flowcharts of linear Transformer encoder (a) and linear attention mechanism (b).

Figure 3a represents the encoder structure, F^A denotes frame-level features extracted from anchor videos, and $F^R \in (F^P, F^N)$ refers to frame-level features from positive and negative sample videos. The model first encodes positional and temporal information of input features, then passes through MHA, layer normalization, and feed-forward network layers. Figure 3b illustrates the linear attention mechanism used in the multi-head attention layer. $\phi(\cdot)$ and $\mathcal{R}_{(\cdot)}$ denote the replacement functions and rotational positional encoding methods. Specifically, $\phi(\cdot) = \text{elu}(\cdot) + 1$ in [24] and $\mathcal{R}_{(\cdot)}$ denotes rotary matrix in [22]. This transformation enhances model training efficiency by focusing on complexity in terms of feature dimensions.

The input video is divided into three categories. Anchor refers to the target query video. Positive is similar in content to the anchor video. Negative is completely different from the anchor video. Anchor samples define the benchmark. Positive samples attract the training model to make same-category samples closer in feature space. Negative samples push the training model to make different-category samples more dispersed in feature space. During model training, the similarity between anchor and positive pairs is maximized. Simultaneously, the similarity between anchor and negative pairs is minimized.

The extracted video features undergo further feature reshaping, where a 2D feature matrix of dimensions $9 \times d$ is concatenated along the feature dimension to form a single $9 \times d$ feature vector representing the feature vector of a frame, where d represents the feature dimension of a frame, in this study, we set it to 512. This process yields the frame-level feature representation. To embed positional information into the frame-level feature vectors, positional encoding based on the sine and cosine functions is utilized.

To obtain a global feature vector representing the entire video, a learnable feature is embedded at the beginning of each video's frame-level feature sequence. When the model converges, this value serves as the spatiotemporal global feature of the given video. Specifically, the input features for attention calculation are as follows:

$$F_{\text{input}}^{A, P, N} = [f_0^{A, P, N}, f_1^{A, P, N}, \dots, f_n^{A, P, N}] + f_{\text{temp}} \quad (1)$$

The f_n^A , f_n^P and f_n^N respectively serve as spatiotemporal global learnable feature representations embedded for anchor, positive sample and negative sample videos, where n indicates the number of frames. The $f_{\text{temp}} = PE(f)$ is obtained by the sine and cosine function PE as proposed in [25]. $F \in (F_{\text{input}}^A, F_{\text{input}}^P, F_{\text{input}}^N)$ simplifies the input video feature used for computing the attention mechanism. Next, learnable parameter matrices W^Q , W^K , W^V are used to transform the input video features F through three different linear transformations into Query Q , Key K and Value V as follows:

$$\begin{aligned} Q &= F^T W^Q \\ K &= F^T W^K \\ V &= F^T W^V \end{aligned} \quad (2)$$

The self-attention layer focuses solely on dynamic changes within individual sequences of video frames, where the input variables Q , K and V are derived from the same feature sequence. In contrast, for the cross-attention module, the K and V remain consistent but differ from the Q . This module is used to learn the interdependencies or similarities between different sequences of video features. Additionally, similar to the MHA in Transformer for parallel computation, this paper employs nine attention heads. These heads redistribute concatenated features into nine distinct feature subspaces, corresponding to extracted features from nine regions. This calculates feature weight relationships across different regions.

When calculating the feature similarity between frames in a video, it is important to consider the varying importance of features across different regions within a frame. This study focuses on nine regions within each frame. To simulate the contribution of these different regions to overall similarity, a weighted approach is adopted. A trainable fully connected layer and a linear layer are employed to generate corresponding weight values. These weights are used to adjust the significance of each region's feature. This approach effectively captures the similarity in content between different region features, thereby more accurately assessing the similarity between frames.

Global spatiotemporal features are subjected to ℓ_2 normalization and a multi-layer perceptron classifier with a single hidden layer to obtain predicted overall video similarity p^{AR} .

This paper adopts a serial stacking of self-attention and cross-attention layers once each. In this mode, the self-attention layer first captures intra-frame similarities within input videos. This is followed by cross-attention, which computes inter-frame features across different videos.

3.4. Similarity Calculation and Loss Function Design

In this paper, the CS is used to compute frame-level similarity of videos and integrating it with CNN models, and we effectively derive similarity scores between videos. This approach efficiently utilizes region feature information to capture temporal similarity patterns between matched frame sequences. For two given videos with frame feature sequences represented as $x = [x_0, x_1, \dots, x_{m-1}]^T$ and $y = [y_0, y_1, \dots, y_{m-1}]^T$, where n, m correspond to the number of frames, and the similarity can be achieved as follows:

$$CS(x, y) = \frac{1}{nm} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \max_{j \in [1, m]} x_i y_j^T \quad (3)$$

It uses CS to compute the average similarity of each element in video sequence x with the most similar item in video sequence y . We calculate similarity scores between frames within videos and between different videos with region features. A Convolutional Attention Neural Network (CANN), which incorporates coordinate attention (CA), is employed to aggregate these similarity values, as illustrated in Figure 4.

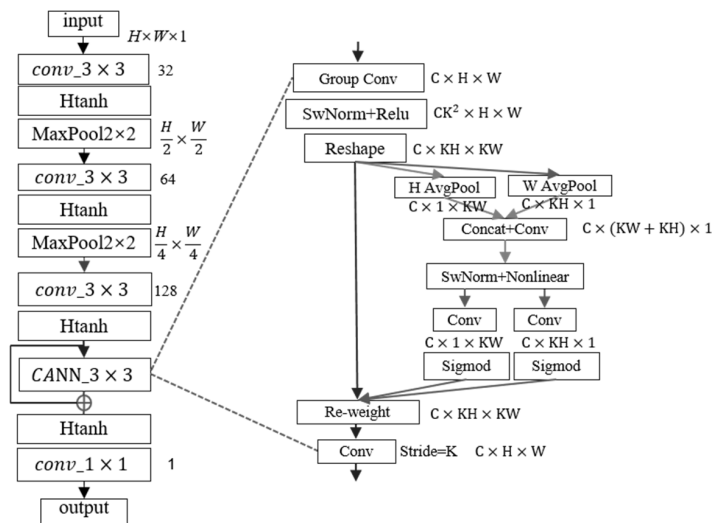


Figure 4. Structure of similarity aggregation network.

Convolutional operations are performed on the similarity score maps between input videos. To better capture both sharply peaked high values and smoothly varying continuous values in the similarity maps, a group convolution [26] incorporating CA is used. This technique is applied between the third and fourth layers of regular convolutions. This approach reduces parameter sharing among convolution kernels and focuses more on the range of receptive fields to enhance the robustness of similarity aggregation. Subsequently, Switchable Normalization (SwNorm) [27] and the Htanh activation function are employed with residual connections to optimize model training, facilitating improved modeling of temporal video similarity.

Use L_1 loss function to calculate the global loss value denoted as $\mathcal{L}_{glob}$ in Equation (4) using the labeled values 1 and -1 for weakly supervised training with the global similarity

score $p^{AR} \in (p^{AP}, p^{AN})$. The 1 and -1 are used to represent positive (P) and negative (N) samples. Due to the response truncation range of the $Htanh$ activation function, it responds to changes within the input domain $[-1, 1]$. A constrained loss $\mathcal{L}_{lim}$ is used to aggregate and sum the inputs of the activation function $\mathcal{M}_v^{xy}$. These inputs are aggregated through the convolutional network. This process computes similar values that fall outside the truncation range. The goal is to map the entire model's calculation of video feature similarity scores to within $[-1, 1]$. This aligns with the common cosine similarity calculation method which outputs values within $[-1, 1]$.

$$\mathcal{L}_{glob} = L_1(1 - p^{AP}) + L_1((-1) - p^{AN}) \quad (4)$$

$$\mathcal{L}_{lim} = \sum_{i=1}^{N/4} \sum_{j=1}^{M/4} |max\{0, \mathcal{M}_v^{xy}(i, j) - 1\}| + |min\{0, \mathcal{M}_v^{xy}(i, j) + 1\}| \quad (5)$$

For the input positive samples, negative samples and anchor videos, we employ a triplet loss function $\mathcal{L}_{trp}$ to compute the difference in similarity loss between the frame-level features of positive sample pairs and negative sample pairs. Through model training, the similarity $\mathcal{G}(\mathbf{F}^A, \mathbf{F}^P)$ of positive sample pairs is increased and the similarity $\mathcal{G}(\mathbf{F}^A, \mathbf{F}^N)$ of negative sample pairs is decreased, thereby minimizing the distance for videos with similar content and increasing it for videos with dissimilar content. The parameter γ adjusts the model for effective training.

$$\mathcal{L}_{trp} = max\{0, \mathcal{G}(\mathbf{F}^A, \mathbf{F}^P) - \mathcal{G}(\mathbf{F}^A, \mathbf{F}^N) + \gamma\} \quad (6)$$

$$\mathcal{L}_{reg} = Smooth - L_1\{\mathcal{G}(\mathbf{F}^A, \mathbf{F}^R), \mathcal{G}^{label}(\mathbf{F}^A, \mathbf{F}^R)\} \quad (7)$$

Triplet loss calculation requires effective construction of positive and negative sample pairs for efficient model training. To optimize the network, higher similarity calculations are assigned to positive video pairs and lower similarity to negative pairs. Pre-obtained similarity values between videos serve as supervision labels $\mathcal{G}^{label}(\mathbf{F}^A, \mathbf{F}^R)$, used for computing region feature similarity loss information, represented as $\mathcal{L}_{reg}$ with $\mathbf{F}^R \in (\mathbf{F}^P, \mathbf{F}^N)$.

A $Smooth - L_1(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$ function is chosen as the loss function for stable loss computation, ensuring smoother curves within $[-1, 1]$ compared to the L_1 loss function. The total loss function is expressed as below, where r is a parameter controlling the impact of similarity penalty loss on the total loss.

$$\mathcal{L} = \mathcal{L}_{glob} + \mathcal{L}_{trp} + \mathcal{L}_{reg} + r * \mathcal{L}_{lim} \quad (8)$$

After the model converges, the parameters are saved for video copyright detection. First, the input video pairs are preprocessed by extracting video frame sequences, padding frame counts for consistency and cropping frame images to the same resolution. Then, video region features are extracted. Next, the model calculates the similarity between different video features. It compares the obtained similarity score with a preset plagiarism similarity threshold. Finally, the model outputs a result indicating whether the video contains plagiarized content.

4. Experimental Results and Analysis

4.1. Experimental Environment and Dataset

Firstly, let us introduce the data used in this study. The video datasets used for training and evaluation are listed in Table 1. The video content in the datasets simulates real-life

video edits involving copyright issues such as video clipping, rotation, image fusion and color adjustment. Therefore, the content of the retrieval video set is considered a type of copyright detection dataset. The CC_WEB_VIDEO dataset is used for the near-duplicate video retrieval task. It contains 24 query video sets and 13,112 related videos. These video samples were obtained by submitting 24 popular text queries to popular video services like YouTube. The EVVE dataset is designed for the event video retrieval task. It contains 2375 related videos and 620 query videos. The main retrieval goal is to capture all videos depicting the event described in the query video. The dataset was collected by using 13 major events as queries provided to YouTube. The VCDB dataset is also sourced from popular video service platforms like YouTube. It serves as a benchmark dataset for partial video copy detection. It contains two subsets: the core subset and the distractor subset. The core subset contains 28 categories with 528 videos, including over 9000 pairs of copied segments. The distractor subset consists of about 100,000 randomly collected videos used as background noise.

Table 1. The video datasets used for training and evaluation.

Dataset	CC_WEB VIDEO [28]	EVVE [29]	VCDB [30]	FIVR-200K [31]	SVD [32]	DnS-100K [16]
Query	24	620	528	100	1206	21,997
Videos	12,790	102,375	100,528	225,960	562,013	115,792
Average Seconds	151.02	194.67	72.77	113.12	17.33	/

The FIVR-200K dataset is used as a benchmark for Fine-Grained Incident Video Retrieval (FIVR). It includes 225,960 videos collected based on 4687 events and 100 video queries. This dataset contains video-level annotation tags involving near-duplicate, duplicate scene, complementary scene and incident scene. The SVD dataset is used for short-video duplicate-content retrieval tasks. It includes 562,013 short videos obtained from a large short-video-sharing platform named Douyin. This dataset includes 1206 query videos, 34,020 tagged video pairs, and 526,787 untagged videos. The average length of the videos is 17.33 s, and the video content is closer to real-life scenarios. This dataset DnS-100K comprises 115,792 videos collected similarly to FIVR-200K but with a shorter time range. FIVR-5K [11] was used as an evaluation dataset to validate model similarity detection performance. It is a subset of FIVR-200K consisting of 50 query videos and 5000 reference videos, divided into three subtasks: DSVR, CSVr, ISVR.

All video feature sequences are truncated to a maximum length of 300 frames. Additionally, they are truncated to a minimum length of 5 frames. All frames within the same batch are aligned to the longest video frame count, with zero-padding applied to shorter sequences to ensure consistency in capturing temporal structure within the similarity matrix. The backbone feature extraction network is ResNet50. The parameter settings for the experiment are shown in Table 2. The specific experimental environment configuration is shown in Table 3.

Table 2. Experimental configuration parameters.

Parameter	Symbol	Parameter Values
Batch Size	B	32
Frame Size	H/W	224
Frame Number	N	300
Learning Rate	learning rate	0.0001
Optimizer	optimizer	Adam
Weight Decay	decay	0.0001

Table 2. *Cont.*

Parameter	Symbol	Parameter Values
Loss Function Parameters	γ	1
Loss Function Weight	r	0.1
Data Augmentation Parameters	β	0.5
Similarity Threshold	θ	0.5
Training Rounds	epoch	300

Table 3. Experimental environment configuration.

Configuration	Parameter
Kernel	Ubuntu 5.15.0-52-generic
Processor	12th Gen Intel(R) Core(TM) i9-12900KF (Intel, Santa Clara, CA, USA)
Memory	32 GB
Graphics Card	NVIDIA A6000 48 GB (NVIDIA, Santa Clara, CA, USA)
Development Platform	Miniconda3
Programming Language	Python 3.10
Deep Learning Framework	PyTorch 1.13.1

4.2. Analysis for Training and Similarity Calculation

Figure 5 illustrates the loss function curve on the training dataset DnS-100K and the mAP curve for Duplicate Scene Video Retrieval (DSVR) task on the FIVR-5K validation dataset. As training progresses, the model gradually learns better feature representations. This process leads to a gradual decrease in the loss function. The loss function exhibits fluctuations, but overall, it shows a decreasing trend. After a certain number of training iterations, the loss function stabilizes. This indicates that further training may result in only marginal improvements. If the loss function continues to decrease on the training set but accuracy on the validation set starts to decline, it may indicate overfitting. However, no such phenomenon is observed in the curve, indicating that the model fits well to the video feature and achieves good convergence. Once the model reaches a certain level of training, accuracy stabilizes at a relatively high level. This indicates that the model fits well with new data for accurate computation.

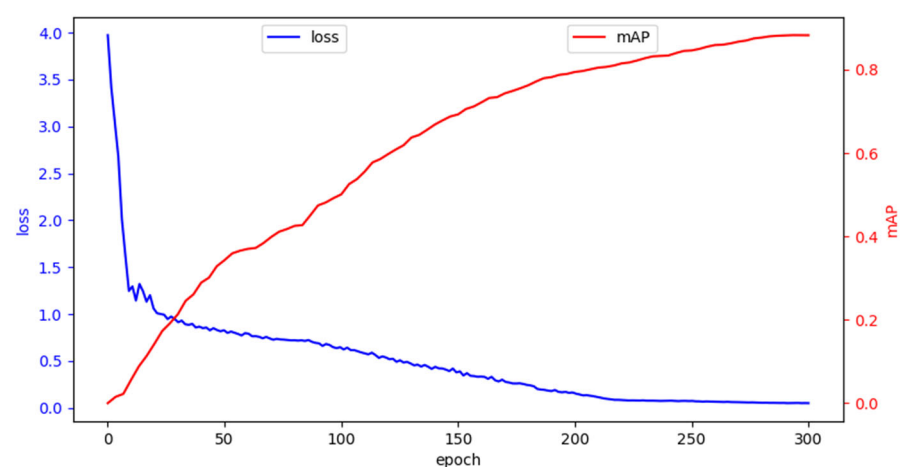
**Figure 5.** Training loss and mAP of validation sets.

Figure 6 illustrates the similarity matrices, aggregating frame-to-frame similarity graphs of query videos using coordinate attention mechanism. Figure 6a shows the heatmap

of frame-to-frame similarity before convolution. A total of 64 frames form a 64×64 similarity matrix. Higher scores are indicated by redder colors. This suggests higher content similarity between the corresponding frames. The entire matrix exhibits a red diagonal representing matching content. There are also areas with lower similarity values around it. They influence the final score. After convolution, the size of the similarity matrix is reduced to one-fourth of its original size. This reduction forms a new heatmap Figure 6b. Through convolution, areas with lower similarity values are filtered out. These areas are represented by deeper blue colors. This process concentrates more similarity calculations along the diagonal. This enhances the accuracy in similarity computation. It focuses more on similar content.

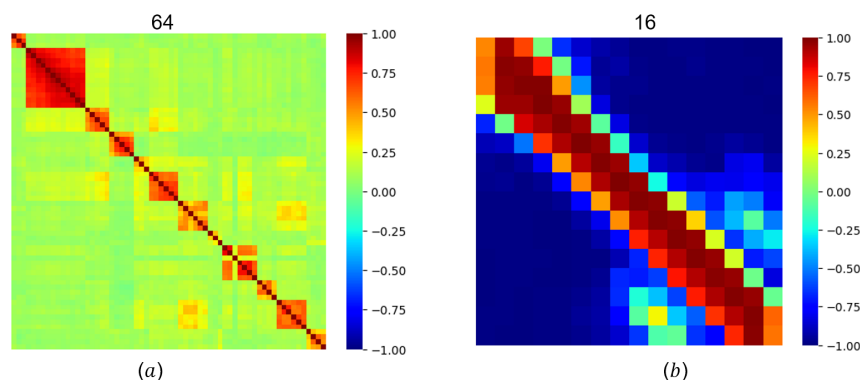


Figure 6. Visualization of similarity plots between frames. Subfigures (a,b) indicate heatmaps of frame-to-frame similarity before and after convolution.

4.3. Comparative Experiment

The mAP values are compared as shown in Table 4. For the DML model, its aggregation of video features into a single feature vector loses a substantial amount of fine-grained feature information. It exhibits relatively lower accuracy performance on the three metrics of the FIVR-5K dataset. Similarly, the TCA_c , TCA_v and DnS_c models, which use video-level features, employ Transformer models based on long sequence modeling networks to fuse feature vectors of video frames. They consider the temporal information from video frames. These models achieve significantly better detection results than the DML model. This validates the effectiveness of using Transformer networks for handling long sequence tasks like short video analysis. The SRA model performs relatively well across the three tasks. It integrates video retrieval and duplicate fragment localization methods. All tasks achieve results above 0.7. Compared to the aforementioned models, the TCA_f and $ViSiL_f$ methods use features of video frame images. They also use local features of frame image regions to compute similarity. These methods effectively consider the temporal variations in videos. They also consider the fine-grained features of various image regions. As a result, they further improve detection accuracy. They achieve a performance gain of over 10 percentage points. These results on the dataset fully illustrate that calculating similarity based on video frame-level features can better detect similar content between videos.

The proposed model in this paper utilizes frame region features as video representations. It achieves detection results of 0.889, 0.878, and 0.801. The DnS_f involves distillation training on the $ViSiL_v$. The S^2VS employs self-supervised learning. Compared to methods like TCA_v and DnS_c that require feature aggregation as input to Transformers, the proposed method can directly use region features as input, combining self-attention and cross-attention for feature enhancement, and extracting global spatiotemporal features to supervise model training. It increases training flexibility.

Table 4. Comparative results of different models on multiple datasets using mAP metric. The best and the second best values are represented in bold and underlined. For the detection of short videos, the proposed method has the best performance.

Method	FIVR-5K			CC WEB VIDEO	EVVE	SVD
	DSVR	CSVR	ISVR			
DML [14]	0.391	0.399	0.380	0.979	0.531	0.785
TCA _c [18]	0.609	0.617	0.578	0.983	0.598	/
DnS _c [16]	0.634	0.647	0.608	0.980	0.636	0.868
SRA [12]	0.700	0.719	0.705	/	/	/
S ² VS [17]	0.881	0.875	0.786	0.996	0.659	0.892
TCA _v [18]	0.726	0.735	0.701	/	/	/
TCA _f [18]	0.844	0.834	0.763	0.994	0.603	/
DnS _f [16]	0.891	0.880	0.802	<u>0.995</u>	0.651	<u>0.902</u>
ViSiL _f [11]	0.856	0.848	0.768	0.993	0.589	0.878
ViSiL _v [11]	0.880	0.869	0.777	0.996	<u>0.658</u>	0.881
Ours	<u>0.889</u>	<u>0.878</u>	<u>0.801</u>	<u>0.995</u>	0.656	0.907

For the CC_WEB_VIDEO dataset, our method achieves similar results as previous methods. For the EVVE and SVD datasets, the detection results are 0.656 and 0.907. The proposed method achieves the best detection result on the real-life short video dataset, demonstrating the effectiveness and generalizability in detecting similarity in videos. This lays the foundation for subsequent video copyright detection with threshold integration. The following sections will verify the necessity and roles of each module used in our method through various ablation experiments.

4.4. Ablation Experiment

The proposed model can be divided into four parts: feature extraction, linear Transformer feature enhancement, similarity computation and loss function design. The feature extraction module, combined with data augmentation and CNN-based similarity aggregation using four-layer convolutions, along with the triplet loss function $\mathcal{L}_{trp}$ and similarity penalty loss function $\mathcal{L}_{lim}$. The feature enhancement module based on attention mechanism, temporal information encoding for video frames, global feature supervision loss $\mathcal{L}_{glob}$, similarity label loss $\mathcal{L}_{reg}$, and group convolution CANN with coordinate attention mechanism and residual connections are used as the ablation targets. The DSVR from the FIVR-5K dataset is used as a validation dataset, with mAP as the evaluation metric. Ablation results are obtained by adding or removing corresponding modules, as detailed in Table 5.

Table 5. Ablation experiment results on different modules.

Ablation Object	✓ Represents Usage						
Backbone	✓	✓	✓	✓	✓	✓	✓
FT		✓	✓		✓	✓	✓
TE			✓		✓	✓	✓
CANN				✓	✓	✓	✓
GF						✓	✓
SM							✓
DSVR	0.765	0.778	0.789	0.771	0.799	0.808	0.889

For the base backbone network, the lack of curated triplet training video pairs with distinct discriminative features led to a predictably low evaluation score of 0.765. Then,

feature enhancement transformations (FT) combined with temporal positional encoding (TE) was used, and score improvements of 0.013 and 0.024 were achieved. Incorporating coordinate attention mechanism into similarity convolution aggregation resulted in a performance improvement of 0.006, indicating that introducing residual connections and attention can aggregate similarity values to some extent while enhancing robustness. The linear Transformer combines self-attention and cross-attention mechanisms. It facilitates interactive information exchange among video features. This enhancement improved the model, enabling it to distinguish similar content videos by 0.034. Finally, integrating global feature (GF) and region similarity measurement (SM) significantly improved accuracy, achieving the highest detection effect of 0.889. Supervised training through weak labels for global features yielded a precision improvement of 0.009, validating the effectiveness of global feature extraction supervision. Using the similarity measurement resulted in a noticeable improvement of 0.081 in accuracy.

We further verified the authenticity of the accuracy improvement brought by the loss functions, as shown in Table 6. The basic backbone network and other module parameters remained unchanged. The similarity values were trained by triplet method. The triplet method carries supervised information. Its purpose is to distinguish positive and negative sample pairs. This allows the model to optimize training with supervised information from another perspective. Higher detection accuracy was achieved in the final detection result. The design of loss function focuses on learning similarity among regional features of videos.

Table 6. Ablation experiment results on different loss functions.

Ablation Object		√ Represents Usage		
$\mathcal{L}_{lim}$	√	√	√	√
$\mathcal{L}_{trp}$	√	√	√	
$\mathcal{L}_{glob}$		√	√	√
$\mathcal{L}_{reg}$			√	√
DSVR	0.799	0.808	0.889	0.861

4.5. Video Copyright Detection Experiment

The proposed model uses a similarity threshold for copyright comparison judgments. Scores above the threshold indicate videos containing plagiarized content, while scores below indicate non-plagiarized content. F1-score and accuracy (Acc) are used as key metrics to comprehensively assess the effectiveness of the detection performance.

The datasets were used to simulate plagiarism using various methods, including geometric transformations like rotation and scaling of video frame images, content overlays, video acceleration, and varying video storage formats. For the CC_WEB_VIDEO and VCDB datasets, this study selects videos closest in category and annotation as original detection videos and considers related videos as replicated content videos, with unrelated videos as non-replicated videos. The FIVR dataset uses ISVR annotations of each query video as indices for replicated videos. EVVE and SVD datasets categorize positive sample videos as replicated videos based on video annotations. Due to different distributions of duplicate content across datasets, an empirical similarity detection threshold $\theta = 0.5$ is set. The results of copyright detection are shown in Table 7.

The proposed model exhibits outstanding performance for the CC_WEB_VIDEO dataset. It achieves a precision of 0.879. The recall and F1 score are both above 0.86. Additionally, the accuracy is 0.931. This indicates the proposed model can efficiently distinguish between plagiarized and non-plagiarized videos. For the FIVR and EVVE

datasets, the performance has decreased a little. For the SVD and VCDB datasets, the proposed model shows stable performance.

Table 7. Copyright detection results on multiple datasets using precision, recall, F1 and accuracy. The best and the second best values are represented in bold and underlined.

Dataset	Precision	Recall	F ₁	Acc
CC WEB	0.879	<u>0.861</u>	0.875	0.931
FIVR	0.710	0.728	0.718	0.774
EVVE	0.582	0.724	0.652	0.729
SVD	<u>0.765</u>	0.890	<u>0.824</u>	<u>0.889</u>
VCDB	0.751	0.816	0.781	0.847
Average	0.737	0.804	0.770	0.834

5. Conclusions

This study proposes a video copyright detection method with region-based feature learning, which computes similarity between videos to determine the presence of copyright infringement. It outperforms frame-level and video-level features in achieving higher detection accuracy. The linear Transformer encoder integrates cross and self-attention mechanisms. This integration enhances feature representation capabilities. The global features effectively supervise model training using global spatio-temporal information. This yields significant improvements in detection performance. By employing Chamfer similarity calculation and aggregating video similarity graphs, we can achieve robust video similarity calculation. Comparative results show that the proposed method achieves the best mAP detection accuracy of 0.907 on the short video dataset SVD. This indicates that the proposed method has certain advantages in detecting real-life short video similarity. Furthermore, combining copyright detection experiments with threshold discrimination, it achieves an average detection precision of 0.737, recall rate of 0.804, F1 value of 0.77, and accuracy of 0.834. Experimental results in video copyright detection demonstrate the effectiveness across multiple datasets in detecting plagiarized videos, affirming its utility and efficacy for practical environment.

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

Funding: This research was funded by the Education Department of Jilin Province, funding number JJKH20250414KJ; the Jilin Provincial Department of Science and Technology (General Program of Jilin Provincial Natural Science Foundation), funding number 20250102243JC; and the National Natural Science Foundation of China, funding number 62062064.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

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

References

1. Tancik, M.; Mildenhall, B.; Ng, R.S. StegaStamp: Invisible hyperlinks in physical photographs. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2114–2123.
2. Liu, Y.; Guo, M.; Zhang, J.; Zhu, Y.; Xie, X. A novel two-stage separable deep learning framework for practical blind watermarking. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1509–1517.
3. Luo, X.; Zhan, R.; Chang, H.; Yang, F.; Milanfar, P. Distortion agnostic deep watermarking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13548–13557.

4. Ahmadi, M.; Norouzi, A.; Karimi, N.; Samavi, S.; Emami, A. ReDMark: Framework for residual diffusion watermarking based on deep networks. *Expert Syst. Appl.* **2020**, *146*, 113157. [[CrossRef](#)]
5. Gao, Y.; Kang, X.; Chen, Y. A robust video zero-watermarking based on deep convolutional neural network and self-organizing map in polar complex exponential transform domain. *Multimed. Tools Appl.* **2021**, *80*, 6019–6039. [[CrossRef](#)]
6. Luo, X.; Li, Y.; Chang, H.; Liu, C.; Milanfar, P.; Yang, F. Dvmark: A deep multiscale framework for video watermarking. *IEEE Trans. Image Process.* **2023**, *34*, 4371–4385. [[CrossRef](#)] [[PubMed](#)]
7. Hasan, H.R.; Salah, K. Combating deepfake videos using blockchain and smart contracts. *IEEE Access* **2019**, *7*, 41596–41606. [[CrossRef](#)]
8. Wang, L.; Bao, Y.; Li, H.; Fan, X.; Luo, Z. Compact CNN based video representation for efficient video copy detection. In Proceedings of the 2016 International Conference on Multimedia Modeling, Miami, FL, USA, 4–6 January 2016; pp. 576–587.
9. Nie, X.; Jing, W.; Ma, L.Y.; Cui, C.; Yin, Y. Two-layer video fingerprinting strategy for near-duplicate video detection. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 555–560.
10. Han, Z.; He, X.; Tang, M.; Lv, Y. Video similarity and alignment learning on partial video copy detection. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 4165–4173.
11. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, I. Visil: Fine-grained spatio-temporal video similarity learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6351–6360.
12. Jo, W.; Lim, G.; Hwang, Y.; Lee, G.; Kim, J.; Yun, J.; Jung, J.; Choi, Y. Simultaneous video retrieval and alignment. *IEEE Access* **2023**, *11*, 28466–28478. [[CrossRef](#)]
13. He, X.; Pan, Y.; Tang, M.; Lv, Y. Self-supervised video retrieval transformer network. *arXiv* **2021**, arXiv:2104.07993.
14. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, Y. Near-duplicate video retrieval with deep metric learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 347–356.
15. Toliás, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2016**, arXiv:1511.05879.
16. Kordopatis-Zilos, G.; Tzelepis, C.; Papadopoulos, S.; Kompatsiaris, I.; Patras, I. DnS: Distill-and-select for efficient and accurate video indexing and retrieval. *Int. J. Comput. Vis.* **2022**, *130*, 2385–2407. [[CrossRef](#)]
17. Kordopatis-Zilos, G.; Toliás, G.; Tzelepis, C.; Kompatsiaris, I.; Patras, I.; Papadopoulos, S. Self-supervised video similarity learning. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 4756–4766.
18. Shao, J.; Wen, X.; Zhao, B.; Xue, X. Temporal context aggregation for video retrieval with contrastive learning. In Proceedings of the 2021 IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 3268–3278.
19. He, S.; He, Y.; Lu, M.; Jiang, C.; Yang, X.; Qian, F.; Zhang, X.; Yang, L.; Zhang, J. TransVCL: Attention-enhanced video copy localization network with flexible supervision. In Proceedings of the 2023 AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 799–807.
20. Chou, C.-L.; Chen, H.-T.; Lee, S.-Y. Pattern-based near-duplicate video retrieval and localization on web-scale videos. *IEEE Trans. Multimed.* **2015**, *17*, 382–395. [[CrossRef](#)]
21. Tan, H.-K.; Ngo, C.-W.; Hong, R.; Chua, T.-S. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In Proceedings of the 17th ACM International Conference on Multimedia, Beijing, China, 19–24 October 2009; pp. 145–154.
22. Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **2024**, *568*, 127063. [[CrossRef](#)]
23. Jégou, H.; Chum, O. Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In Proceedings of the 2012 European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 774–787.
24. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-free local feature matching with transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–25 June 2021; pp. 8922–8931.
25. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *Neural Inf. Process. Systems.* **2017**, *30*.
26. Zhang, X.; Liu, C.; Yang, D.; Song, T.; Ye, Y.; Li, K.; Song, Y. RFACnv: Innovating spatial attention and standard convolutional operation. *arXiv* **2024**, arXiv:2304.03198.
27. Luo, P.; Ren, J.; Peng, Z.; Zhang, R.; Li, J. Differentiable learning-to-normalize via switchable normalization. *arXiv* **2019**, arXiv:1806.10779 2018.
28. Wu, X.; Hauptmann, A.G.; Ngo, C.-W. Practical elimination of near-duplicates from web video search. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 25–29 September 2007; pp. 218–227.

29. Revaud, J.; Douze, M.; Schmid, C.; Jégou, H. Event retrieval in large video collections with circulant temporal encoding. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2459–2466.
30. Jiang, Y.-G.; Jiang, Y.; Wang, J. VCDB: A large-scale database for partial copy detection in videos. In Proceedings of the 2014 European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 357–371.
31. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, I. FIVR: Fine-grained incident video retrieval. *IEEE Trans. Multimed.* **2019**, *21*, 2638–2652. [[CrossRef](#)]
32. Jiang, Q.-Y.; He, Y.; Li, G.; Lin, J.; Li, L.; Li, W.-J. SVD: A large-scale short video dataset for near-duplicate video retrieval. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5281–5289.

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.