Video Copy Detection Using Spatio-Temporal CNN Features

Zhili Zhou¹, Member, IEEE, Jingcheng Chen¹, Ching-Nung Yang², Senior Member, IEEE, and Xingming Sun³, Senior Member

¹Jiangsu Engineering Center of Network Monitoring and School of Computer and Software, NanJing University of Information Science and Technology, Jiangsu 210044 China
²Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan 97401 R.O.C

ABSTRACT To protect the copyright of digital videos, video copy detection has become a hot topic in the field of digital copyright protection. Since a video sequence generally contains a large amount of data, to achieve an efficient and effective copy detection, the key issue is to extract compact and discriminative video features. To this end, we propose a video copy detection scheme using spatio-temporal Convolutional Neural Network (CNN) features. First, we divide each video sequence into multiple video clips and sample the frames of each video clip. Second, the sampled frames of each video clip are fed into a pre-trained CNN model to generate the corresponding convolutional feature maps (CFMs). Third, based on the generated CFMs, we extract the CNN features on the spatial and temporal domains of each video clip, i.e., the spatio-temporal CNN features. Finally, video copy detection is efficiently and effectively implemented based on the extracted spatio-temporal CNN features. Experiments on the commonly used video dataset, i.e., TRECVID 2008, demonstrate that the proposed method performs well in aspects of both accuracy and efficiency and shows superiority to several other copy detection methods using the state-of-the-art features.

INDEX TERMS Video copy detection, video security, copyright protection, spatio-temporal, convolutional feature maps, convolutional neural network (CNN).

I. INTRODUCTION

With the rapid development of multimedia technology and the wide use of Internet, the number of digital video files increase exponentially on the Internet. Also, due to the increasing popularity of various video processing tools, it is convenient for users to replicate and edit videos with a variety of modifications such as cropping, blurring, noise addition, picture in picture, text insertion, and recompression. Thus, there are a lot of video copies distributed on the Internet. Fig. 1 shows several examples of video copies generated by various modifications. To prevent unauthorized use of video content, the protection of digital video copyright has become an urgent problem, and detecting video copies is a basic requirement for video copyright protection.

There are two ways for detecting illegal copies: digital watermarking and content-based copy detection [1]-[4]. Digital watermarking technique needs to embed watermark into the video file before its distribution. All copies of the video files will contain the same watermark, which can be extracted to prove the ownership. Different from watermarking technique, content-based copy detection directly extracts content-based features from a video as its unique information for copy detection. Specifically, a video copy detection system usually collects millions of data crawled from the Internet, and then compares the features extracted from an original video and database videos to determine whether the database videos are copy versions of the original. Compared to the watermarking, content-based copy detection does not require additional information but video itself, and copy detection can be performed after video distribution. In addition, content-based copy detection techniques can be also applied in many other emerging applications [5]-[14].
Most of the existing content-based copy detection methods can be roughly categorized into handcrafted feature-based [15]-[19] and convolutional neural network (CNN) feature-based methods [20]-[24]. One of most typical hand-crafted features used for copy detection is high-dimensional scale invariant feature transform (SIFT) feature [25], which is usually combine with bag-of-visual-word (BOW) model to build an inverted index structure to speed up the feature matching process between videos. Nonetheless, as a short video sequence generally contains a large amount of data and thus a huge number of high-dimensional SIFT features are extracted from the video, the computational complexity of the existing handcrafted feature-based methods is still quite high. Moreover, the handcrafted features can only describe low-level characteristics of videos such as texture information and ignore the video semantic information, which makes the features not discriminative enough and thus limits the detection performance of these methods.

In recent years, with the rapid development of CNN networks, CNN features have already surpassed the traditional handcrafted features in many computer vision tasks. Thus, some CNN feature-based copy detection methods are accordingly proposed [20]-[24] to directly use the output of the CNN networks’ fully connected layers or convolutional layers as video features. However, these CNN features are extracted at frame-level and are still not compact enough. Moreover, most of these features do not sufficiently encode the important temporal information between video frames, which leads to inferior performance.

To efficiently and effectively detect video copies, the key issues is to extract compact and discriminative video features. To this end, we propose a novel video copy detection using spatio-temporal Convolutional Neural Network (CNN) features. Our main contributions are given as follows.

1) After dividing each video sequence into multiple clips and sparsely sampling the frames of video clips, we generate convolutional feature maps (CFMs) from these sampled video frames. These provide foundation for extracting compact and discriminative features.

2) By using the CFM of fused frames of each video clip, we extract 256-dimensional spatial CNN feature, which has high compactness.

3) With the consideration of the temporal characteristics of each video clip, two temporal features are extracted among the CFMs of frames. The two features are the complement to the spatial CNN feature and can help improve the discriminability of features, leading to the improvement of detection performance.

The rest of this paper is organized as follows. In Section II, we introduce the related works. The details of the proposed copy detection method are given in Section III. Section IV gives the experimental results and analysis, and Section V draws conclusions.

II. Related works

Generally, the existing video copy detection methods use handcrafted features or CNN features. In earlier work, video copy detection methods mostly use handcrafted features including SIFT [25] and its extensions such as principal component analysis on SIFT (PCA-SIFT) feature [26] and speeded-up robust feature (SURF) [27].

Kim et al. [15] compute the average intensities of blocks of each video frame and extract the ordinal signatures by sorting these intensity values for video copy detection. Douze et al. [16] extract SIFT features from video frames and match individual video frames based on their SIFT features and the BOW model to detect video copies. In [17], Özbulak et al. combine SURF features with ORB features for copy detection. Specifically, each frame in a video is represented by a set of SURF keypoints, and these points are then described by ORB descriptors, which are represented as a set of 32-dimensional feature vectors. In [18], Yang et al. propose a robust hashing algorithm based on SURF and ordinal measure. First, they extract SUFT features in a frame-by-frame manner and divide each frame into $4 \times 4$ blocks. Then, each block is traversed by Hilbert-order rasterization to count the number of SURF points to generate video signature. Lei et al. [19] split each keyframe into nonoverlapping blocks, and sort the color components of each block according to their average intensities to generate the color correlation features. Then, the generated color correlation features are matched between frames for video copy detection.

Recently, the convolutional neural networks (CNN) have gained great success in the field of computer vision. In view of the outstanding performances in image classification and object recognition, some researches also extent CNN to the area of video copy detection/video retrieval and show that the features based on CNN generally outperform traditional handcrafted features. Some methods obtain representations of video frames directly from the fully connected layers of CNN,
while others extract more stable descriptors from the output of the convolutional layers.

Liu et al. [20] use the CNN model to detect object regions from video frames, and then these regions are used to generate binary fingerprints for fast copy detection. Li et al. [21] use a 3D-CNN model to extract features from video streams directly, and then convert multi-classification problems into multiple bi-classification problems for copy detection. Lou et al. [22] propose a Nested Invariance Pooling (NIP) method to obtain compact and robust CNN descriptors. Specifically, the CNN descriptors are generated by applying three different pooling operations on the output of last convolution layer of CNN with input video frames. Kordopatis-Zilos et al. [23] extract frame-level feature descriptors by applying max pooling on the activations of multiple convolutional layers of CNN. Zhang et al. [24] directly use the famous CNN model, i.e., Alexnet's output of the sixth full connected layer (FC6-layer) as the keyframe-level representation, and then use an exhaustive search-based matching method to retrieve video copies from databases.

III. The Proposed Video Copy Detection Method

This section gives the details of the proposed method. Fig. 2 shows the framework of the proposed copy detection method. First, for a video, we divide it into multiple video clips. Then, we sparsely sample the frames of each video clip to obtain the sampled frames. These sampled frames are fed into a pre-trained CNN model to extract spatial-temporal CNN features. Also, we build the inverted index file based on the extracted spatial-temporal CNN features with the BOW model to efficiently and effectively detect copy versions of the query from databases. The details of the proposed method are described as follows.

A. Grouping and Sampling Video Frames

A video sequence is typically composed of a large number of video frames, and thus directly extracting features from these frames will have very high computational complexity. Therefore, in this paper, we group a video sequence into a set of multiple video clips and then sample the frames in the video clips to extract features at video clip-level. Suppose a video sequence $V$ contains $N$ video frames. Then we divide the video sequence into a set of $\lceil N/L \rceil$ video clips denoted as $C_i = \{s(1), s(2), \ldots, s(T)\}$, where $L$ represents the number of video frames per video clip. Then, each video clip $C_i$ is sampled to $S_i = \{s(i,1), s(i,2), \ldots, s(i,T)\}$ with the sampling interval $b$, where $T = \lfloor L/b \rfloor$ means the number of frames in the sampled video clip $S_i$.

B. Generation of Spatial CNN Features

For a sampled video clip $S_i$, all of its frames can be sequentially fed into a CNN model for feature extraction. However, that will make the computational complexity very high. In order to reduce the computational complexity of feature extraction, we use formula (1) to fuse all the frames in each samples video clip $S_i$ into a single image.

\[
I_i = \frac{\sum_{j=1}^{T} s(i,j)}{T}
\] (1)
Where \( I_i \) represents the fused image generated from sampled video clip \( S_i \).

Generally, CNN features are extracted from the fully-connected layers or the convolutional layers of the pretrained CNN model with input images or video frames. Several methods [28]-[31] have demonstrated that the features extracted from the last convolutional layer show superior performance than the features from the fully connected layers. Therefore, in this paper, the fused image is fed into the pre-training network model, \( \text{i.e.} \), the famous Alexnet network [32], and the output values of the last convolution layer called as convolutional feature maps (CFMs) are obtained to generate the spatial CNN features. The CFMs can be regarded as a set of \( K \) feature maps with the size of \( W \times H \), denoted as \( \text{CFMs}_k = \{M(i,1), M(i,2), \ldots, M(i,K)\} \). Where, \( K \) is equal to 256, and \( W \) and \( H \) are proportional to the width and height of the input fused image [1]-[4]. After that, the sum-pooling operation [33] is implemented on each generated CFM to generate the spatial CNN feature of the video clip \( S_i \), denoted as \( \text{SF}_i = \{s_f(i,1), s_f(i,2), \ldots, s_f(i,K)\} \), \( K = 256 \). Where, the sum-pooling operation is the computation of the sum of all values in each CFM. Fig. 3 shows the flow chart of the extraction of spatial CNN feature from a video clip.

### C. Generation of temporal CNN features

Since a video sequence is arranged by multiple frames in a certain order, its temporal characteristics also play an important role for identifying whether the database video a copy version of a given query.

In order to enhance the discriminability of copy detection system, we also capture the temporal characteristics of the video clips to extract the temporal CNN features. For each video frame in the sampled video clip \( S_i \), we feed it into the CNN model and obtain the corresponding CFMs \( \text{CFMs}_k = \{M(i,1), M(i,2), \ldots, M(i,K)\} \). Then, we use Eq. (2) to compute average CFM of all the CFMs.

\[
\text{CFM}_i = \frac{1}{K} \sum_{j=1}^{K} M(i,j)
\]

Where the size \( \text{CFM}_i \) is same as that of \( M(i,j) \), \( M(i,j) \in \mathbb{R}^{W \times H} \). Denote the centroid of \( \text{CFM}_i \) as \( \text{P}_i \), and the coordinates of \( \text{P}_i \) denoted as \( (X_i,Y_i) \) are computed by

\[
(X_i,Y_i) = \frac{1}{K} \sum_{j=1}^{K} (X_j,Y_j)
\]

As a result, for a video clip \( S_i \), we obtain a set of centroids and its coordinates corresponding to its frames, \( \text{i.e.}, P_i = \{(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}), \ldots, (X_{iK}, Y_{iK})\} \), where \( P_i = (X_i, Y_i) \). Fig. 4 shows the computation process for centroid coordinates of each video frame based on its CFMs.

Next, according to the computed centroid coordinates, we extract two temporal features of the video clip \( S_i \). As shown in Fig. 5, for any two centroids \( P_i \) and \( P_j \) in the centroid set \( P \), we measure two relationships between them, \( \text{i.e.}, \) the distance \( d(i,j) \) and the angle \( \theta(i,j) \). The computation of \( d(i,j) \) and \( \theta(i,j) \) is given as follows.

\[
d(i,j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}
\]

\[
\theta(i,j) = \arctan\left(\frac{Y_j - Y_i}{X_j - X_i}\right)
\]

After computing the distance and angle relationships between the centroids of any two frames in the video clip \( S_i \),
we can obtain two $D = T \times (T-1)/2$ dimensional temporal features, which are denoted as $TDF_i = \{ tdf(i, 1), tdf(i, 2), \ldots, tdf(i, D) \}$ and $TAF_i = \{ taf(i, 1), taf(i, 2), \ldots, taf(i, D) \}$, respectively.

**D. Index construction**

A well-designed index structure is the key to achieve an efficient copy detection. Thus, in this paper, we build the inverted index structure based on the extracted CNN features and the BOW model. More details are given as follows.

We first quantize the spatial features extracted from all video clips to a set of clusters by K-means algorithm, and each cluster is treated as a visual word. Then, each visual word is followed by the IDs of the video clips quantized to the visual word, the spatial CNN features, and the set of centroid coordinates of the video clip frames. It is worth noting that, to save the memory consumption, we do not directly preserve the temporal features in the index file. Instead, the computed centroid coordinates are stored and the temporal features can be efficiently extracted from these coordinates at the online copy detection stage.

Fig. 6 illustrates the structure of inverted index structure used in our copy detection method.

**E. Copy detection**

After the above steps, we obtain the spatial and temporal features of each video clip and build the inverted index file. Next, we will introduce how to implement the copy video detection in detail.

In Section III-D, during the construction of inverted index file, the similar spatial features can be quantized into a same visual word. Thus, for a given query video clip, we first extract its spatial feature by the same spatial extraction algorithm described in Section III-B. Then, by looking up the index file, we can efficiently detect the IDs of suspicious video clips from video databases. Next, we can only check whether these suspicious video clips are the copies of the query video clip by computing their similarity. As a result, the search scope can be narrowed largely.

Denote the spatial features of the query video clip and that of a suspicious clip as $SF_Q$ and $SF_S$, respectively. Similarly, their two temporal features are denoted as $(TDF_Q, TAF_Q)$ and $(TDF_S, TAF_S)$, respectively. Then, we calculate the similarity between the query video clip and the suspicious clip using their spatial and temporal features. The spatial similarity between the spatial features, i.e., $SF_Q$ and $SF_S$, is computed by

$$SIM_s = 1 - \frac{\sum_{i=1}^{K} \max(|SF_Q(i)|, |SF_S(i)|)}{\sum_{i=1}^{K} |SF_Q(i) - SF_S(i)|}$$

**(7)**

Similarly, we compute the similarity between the distance temporal features, i.e., $TDF_Q$ and $TDF_S$, and that between the angle temporal features, i.e., $TAF_Q$ and $TAF_S$ by Eqs. (8) and (9), respectively.

$$SIM_{TD} = 1 - \frac{\sum_{i=1}^{D} \max(|SF_Q(i)|, |SF_S(i)|)}{\sum_{i=1}^{D} |SF_Q(i) - SF_S(i)|}$$

**(8)**

$$SIM_{TA} = 1 - \frac{\sum_{i=1}^{D} \max(|SF_Q(i)|, |SF_S(i)|)}{\sum_{i=1}^{D} |SF_Q(i) - SF_S(i)|}$$

**(9)**

By using $SIM_{TD}$ and $SIM_{TA}$, the temporal similarity can be measured by

$$SIM_T = \alpha SIM_{TD} + (1-\alpha)SIM_{TA}$$

**(10)**

Where, $\alpha$ is a weighting factor ranging from 0 to 1. The final similarity between the query video clip and a suspicious clip is computed by

$$SIM_C = \beta SIM_s + (1-\beta)SIM_T$$

**(11)**

Where, the range of the weighting factor $\beta$ is from 0 to 1. Then, by comparing the computed similarity $SIM_C$ with a predefined threshold $SIM_{TD}$, we can determine whether the suspicious video clip is a copy version of the query.
IV. Experiments

In this section, we will first introduce the validation data sets and evaluation criteria. Second, the parameter $L$, sampling interval $b$ and weight factors $\alpha$ and $\beta$ used in our method are determined by experiments. Third, the performance of the proposed method is tested and compared with the methods using state-of-the-art features.

A. Datasets and evaluation criteria

The commonly used content-based video copy detection dataset, i.e., TRECVID 2008 [34], is adopted to verify the performance of the proposed method. The dataset consists of 31 original videos and 310 videos (MPEG-4 format) generated by 10 common modifications including cropping, blurring, noise addition, picture-in-picture, and their combinations. The 31 original videos are treated as queries. The Alexnet network is adopted to extract spatio-temporal CNN features in our method. We use the PR curve to measure the performance of our proposed method. The PR curve is defined as follows.

\[ \text{Recall} = \frac{N_{\text{Positive}}}{N_{\text{Total}}} \]  
\[ \text{Precision} = \frac{N_{\text{Positive}}}{N_{\text{Detected}}} \]  

Where $N_{\text{Positive}}$ represents the number of true samples detected, $N_{\text{Total}}$ represents the total number of all samples (positive and negative samples) in the database, and $N_{\text{Detected}}$ represents all detected samples by the detection system. Also, we use the mean Average Precision (mAP) to evaluate the detection performance, which represents the average accuracy at different recall levels.

All experiments are performed on a standard PC (3.2 GHz Core-i5 and 8 GB RAM) with Window 7 X64 system.

B. Parameter selection

In this section, we test the impacts of four important parameters in the proposed method, they are $L$, $b$, $\alpha$, $\beta$, respectively. $L$ means the number of frames per video clip, and $b$ represents the interval of sampling, $\alpha$ and $\beta$ are the weighting factors used for similarity measurement between video clips.

We first fix the parameters $\alpha$ and $\beta$ to the default values (i.e., 0.5 and 0.5, respectively) to test the impacts of parameters $L$ and $b$ on mAP values, and show the results in Fig. 7. From the figure, it is clear that smaller $L$ and $b$ lead to better detection performance. This is mainly because smaller $L$ and $b$ cause more video clips and more frames in each clip, which make the features extracted from the clips more discriminative. However, as shown in Fig. 8, smaller $L$ and $b$ lead to much higher computational complexity and much more time cost. Therefore, to achieve a good tradeoff between accuracy, memory consumption, and time cost, we set $L$ and $b$ to 7 and 10, respectively, which are used in the following experiments.

Then, we evaluate the effects of $\alpha$ and $\beta$ by setting $L$ and $b$ to 7 and 10, respectively. The effects of the parameters $\alpha$ and $\beta$ on the mAP is shown in Fig. 9. As
shown in this figure, as the value of $\beta$ changes from 0 to 1, the mAP value increases significantly and then decreases slowly. The main reason is the two features describe the video characteristics on spatial domain and temporal domain, respectively, and can be complementary to each other to improve the performance. Similarly, too large or too small $\alpha$ leads to inferior performance. According to the two figures, when the value of $\alpha$ is equal to 0.9 and $\beta$ is 0.9, the optimal mAP value is obtained. Thus, we set $\alpha$ and $\beta$ to 0.9 and 0.9, respectively, in the following experiments.

C. Performance evaluation
In this subsection, we draw the Precision and Recall (PR) curve to evaluate the performance of our copy detection method. The PR curve of our method on the TRECVID 2008 database is shown in Fig. 10.

To further illustrate the superiority of our method, we next compare it with several other methods using the state-of-art features. The comparison results are shown in Table I. The "Sum-pooling" and "Max-pooling" in the table mean the methods that use spatial video features generated by the sum-pooling and those by Max-pooling strategies, respectively, with the proposed copy detection framework. In Table I, the average time cost per query is used to measure the time efficiency of different methods, and the memory consumption of the inverted index file is used to measure the space efficiency of different methods. From this table, it is clear that, although our method requires slightly higher memory space and time cost than the two other methods, the mAP value of our method is higher than those of two other methods. This can be explained by the facts that the video features extracted by our method, i.e., spatio-temporal features, are more discriminative than the two other features, but the extracted video features lead to more memory consumption and time cost for video copy detection.

In conclusion, our method achieves better accuracy than the methods using the state-of-the-art features, while maintaining good performance in the aspects of memory consumption and time efficiency. Fig. 11 shows some examples of detection results using our method.

V. CONCLUSION
We have presented a video copy detection method based on two kinds of CNN features, i.e., spatial features and temporal features. The CNN features extracted from the sparsely sampled video frames have high compactness, and they can also describe the characteristics of videos on both spatial and temporal domains, which lead to high discriminability. The experimental results show that the proposed video copy detection method can achieve high performances in the aspects of both efficiency and effectiveness.

REFERENCES
Zhili Zhou received the B.S. degree in communication engineering from Hubei University in 2007, and the M.S. and Ph.D. degrees in computer application from the School of Information Science and Engineering, Hunan University, in 2010 and 2014, respectively.

He is an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. Also, he is currently a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Windsor, Canada. His current research interests include near-duplicate image/video retrieval, image search, image/video copy detection, content-based image retrieval, forensics, and image processing.

Zhilong Shi received the B.E. degree in communication engineering from Hubei University in 2007, and the M.S. and Ph.D. degrees in computer application from the School of Information Science and Technology, China. His current research interests include near-duplicate image/video retrieval, image search, image/video copy detection, and natural language processing.

Ching-Nung Yang received the B.S. and M.S. degrees in telecommunication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1997.

He is currently a Full Professor with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. His research interests include coding theory, information security, and cryptography.

Xingming Sun received his BS in mathematics from Hunan Normal University, China, in 1984; his MS in computing science from Dalian University of Science and Technology, China, in 1988; and his PhD in computer science from Pudan University, China, in 2001. He is currently a professor at the College of Computer and Software, Nanjing University of Information Science and Technology, China. In 2006, he visited the University College London, UK; he was a visiting professor in University of Warwick, UK, between 2008 and 2010. His research interests include network and information security, database security, and natural language processing.

Jingcheng Chen received the B.S. degree in Binhai College of Nanjing University of Information Science & Technology in 2017, China. He is currently working towards the M.S. degree in Nanjing University of Information Science & Technology, China. His research interest includes image retrieval, image/video copy detection, information security.


