IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON RECENT ADVANCES IN VIDEO CODING AND SECURITY

Received June 29, 2019, accepted July 16, 2019, date of publication July 22, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930173

Video Copy Detection Using Spatio-Temporal CNN Features

ZHILI ZHOU[1], (Member, IEEE), JINGCHENG CHEN[1], CHING-NUNG YANG[2], (Senior Member, IEEE), AND XINGMING SUN[1], (Senior Member, IEEE)

[1]Jiangsu Engineering Center of Network Monitoring and School of Computer and Software, Nanjing University of Information Science and Technology, Jiangsu 210044, China
[2]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 97401, Taiwan

Corresponding author: Zhili Zhou (zhou_zhili@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602253, Grant U1836208, Grant U1536206, Grant U1836110, and Grant 61672294, in part by the National Key Research and Development Program of China under Grant 2018YFB1003205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China, and in part by the Ministry of Science and Technology (MOST) through Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan, under Contract 108-2634-F-259-001.

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 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. The 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, convolutional neural network (CNN), convolutional feature maps, spatio-temporal CNN features.

I. INTRODUCTION

With the rapid development of multimedia technology and the wide use of Internet, the number of digital multimedia files increase exponentially on the Internet, and many security issues of multimedia data have arisen [1]–[12]. 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 [13]–[16]. 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 videos 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

The associate editor coordinating the review of this manuscript and approving it for publication was Jianjun Lei.

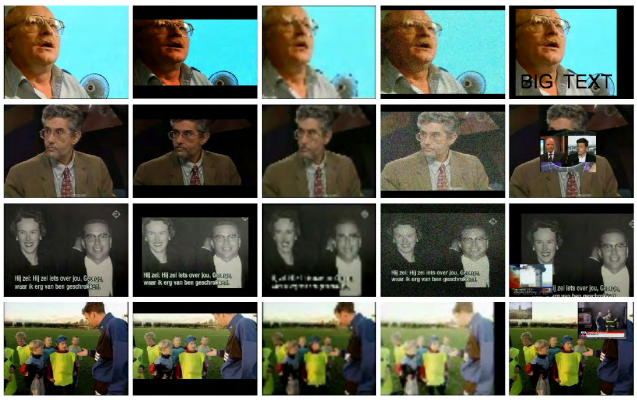


FIGURE 1. The toy examples of video copies. The left column is the original video, and the right four columns correspond to different kinds of copy versions generated by various modifications. They are cropping, blurring, noise addition, and combination of several modifications including text insertion, cropping, noise addition, and picture-in-picture.

additional information but video itself, and copy detection can be performed after video distribution.

Most of the existing content-based copy detection methods can be roughly categorized into handcrafted feature-based [17]–[21] and convolutional neural network (CNN) feature-based methods [22]–[26]. One of most typical hand-crafted features used for copy detection is high-dimensional scale invariant feature transform (SIFT) feature [27], which is usually combined 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 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 make the features not discriminative enough and thus limit 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 [22]–[26] to directly use the output of the CNN model's 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 issue 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 CFMs of the frame fused from 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 promising 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 the conclusions.

II. RELATED WORKS

Generally, the existing video copy detection methods use handcrafted features or CNN features. In earlier works, video copy detection methods mostly use handcrafted features including SIFT [27] and its extensions such as principal component analysis on SIFT (PCA-SIFT) feature [28] and speeded-up robust feature (SURF) [29].

Changick and Vasudev [17] compute the average intensities of blocks divided from each video frame and extract the ordinal signatures by sorting these intensity values for video copy detection. Douze *et al.* [18] 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 [19], Özbulak *et al.* combine SURF features with Oriented Fast and Rotated Brief (ORB) features [30] 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 32-dimensional feature vectors. In [20], Yang *et al.* propose a robust hashing algorithm based on SURF and ordinal measure. First, they extract SURF features in a frame-by-frame manner and divide each frame into 4×4 blocks. Then, each block is traversed by Hilbert-order rasterization to count the number of SURF points to generate video signature. Lei *et al.* [21] 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 researchers also extent CNN to the area of copy detection/retrieval and show that the features based on CNN generally outperform the traditional handcrafted features. Some methods obtain representations of video frames directly from the fully connected layers

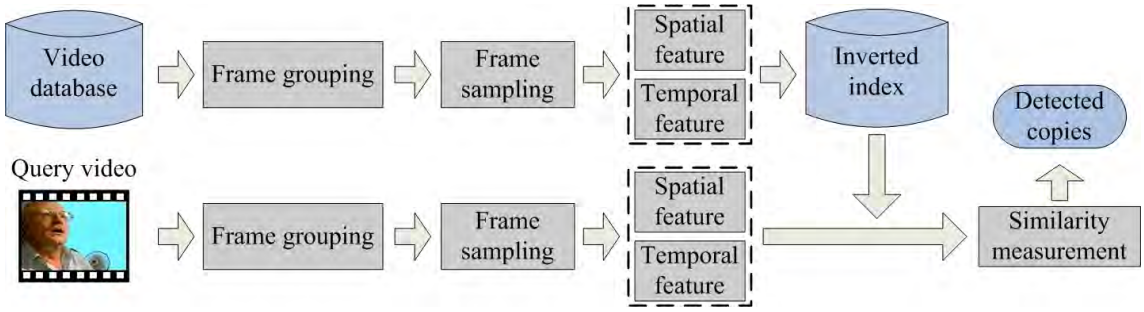


FIGURE 2. The framework of the proposed video copy detection method.

of CNN, while the others extract more stable descriptors from the output of the convolutional layers.

Liu *et al.* [22] 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.* [23] 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*. [24] 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.* [25] extract frame-level descriptors by applying max pooling on the activations of multiple convolutional layers of CNN. Zhang *et al.* [26] directly use the famous CNN model, *i.e.*, Alexnet's output of the sixth full connected (FC6) layer as the keyframe-level representation, and then use an exhaustive search-based matching method to retrieve video copies from databases. Although these CNN features-based methods achieve better detection performance than the traditional handcrafted feature-based methods, they do not sufficiently consider the characteristics of videos on temporal domain, which makes these features not discriminative enough. Moreover, the feature extraction is implemented at frame-level, and thus the compactness of features needs to be further improved.

Therefore, there is still a lot of room for performance improvement for the existing copy detection methods. In this paper, we propose spatio-temporal CNN features for video copy detection. The spatio-temporal CNN features are extracted at video clip-level, which encode the characteristics of video clips on both spatial and temporal domains. Thus, they show high discriminability and compactness. The experimental results demonstrate that the proposed copy detection method based on the spatial-temporal CNN features can efficiently and effectively detect video copies from datasets.

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. Finally, 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 the frames of a video sequence into a set of multiple video clips and then sample the frames in the video clips for the feature extraction at video clip-level. Suppose a video sequence V contains N video frames. Then we divide the video sequence into a set of $\lfloor N/L \rfloor$ video clips denoted as $C = \{C_i | 1 \leq i \leq \lfloor N/L \rfloor\}$, where L represents the number of video frames per video clip. Then, each video clip C_i is sampled to $S_i = \{s(i, 1), s(i, 2), \cdots, s(i, T)\}$ 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 Eq. (1) to fuse all the frames in each samples video clip S_i into a single image.

$$I_i = \frac{\sum_{j=1}^{T} s(i, j)}{T} \tag{1}$$

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 a pretrained CNN model with input images or video frames. Several methods [31]–[34] 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

FIGURE 3. The flow chart of the extraction of spatial CNN feature. First, multiple video frames are fused into an image, and then the CFMs of the image are generated for the spatial CNN feature extraction.

network model, *i.e.*, the famous Alexnet network [35], 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 $CFM_i = \{M(i,1), M(i,2), \cdots, M(i,K)\}$. Where, K is equal to 256, and W and H are proportional to the width and height of the input fused image. Then, the sum-pooling operation [36] is implemented on each generated CFM to generate the spatial CNN feature of the video clip S_i, denoted as $SF_i = \{sf(i,1), sf(i,2), \cdots, sf(i,K)\}$, $K = 256$. 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 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 copy detection.

In order to enhance the discriminability of video features, 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 $CFM_i = \{M(i,1), M(i,2), \cdots, M(i,K)\}$. Then, we use Eq. (2) to compute average CFM of all the CFMs.

$$\overline{CFM_i} = \frac{\sum_{j=1}^{K} M(i,j)}{K} \tag{2}$$

where the size of $\overline{CFM_i}$ is same as that of $M(i,j)$, $M(i,j) \in \mathbb{R}^{W \times H}$. Denote the centroid of $\overline{CFM_i}$ as P_i, and the coordinates of P_i denoted as (X_i, Y_i) are computed by

$$X_i = \frac{\sum_{1 \le x \le W, 1 \le y \le H} x \times \overline{CFM_i}(x,y)}{\sum_{1 \le x \le W, 1 \le y \le H} \overline{CFM_i}(x,y)} \tag{3}$$

$$Y_i = \frac{\sum_{1 \le x \le W, 1 \le y \le H} y \times \overline{CFM_i}(x,y)}{\sum_{1 \le x \le W, 1 \le y \le H} \overline{CFM_i}(x,y)} \tag{4}$$

As a result, for a video clip S_i, we obtain a set of centroids and its coordinates from its frames, *i.e.*, $P = \{P_1, P_2, \cdots, P_T\}$, 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.

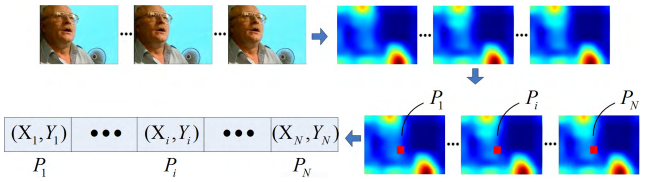


FIGURE 4. The flow chart of the computation of centroid coordinates of each video frame based on its CFMs. First, an average CFM is computed from each video frame's CFMs, and then the coordinates of the centroid are calculated on each average CFM.

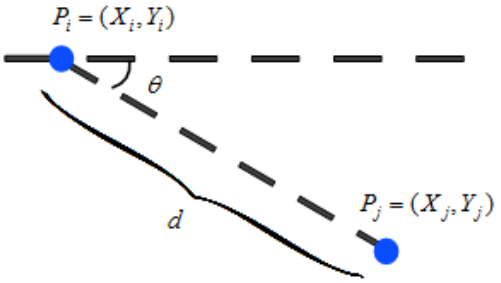


FIGURE 5. The extraction of two temporal features using the centroid coordinates.

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, *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.

$$d(i,j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \tag{5}$$

$$\theta(i,j) = \arctan\left(\frac{Y_i - Y_j}{X_i - X_j}\right) \tag{6}$$

After measuring the distance and angle relationship 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 $TDF_i = \{tdf(i,1), tdf(i,2), \cdots, tdf(i,D)\}$ and $TAF_i = \{taf(i,1), taf(i,2), \cdots, taf(i,D)\}$, 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 CNN 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 recovered from these coordinates at the online copy detection stage.

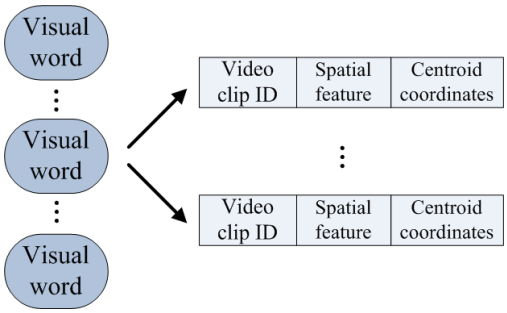


FIGURE 6. The structure of inverted index structure.

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

E. COPY DETECTION

After the above steps, we obtain the spatial and temporal features of each video clips 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

$$SIM_S = 1 - \frac{\sum_{i=1}^{K} |SF_Q(i) - SF_S(i)|}{\sum_{i=1}^{K} \max(|SF_Q(i)|, |SF_S(i)|)} \tag{7}$$

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.

$$SIM_{TD} = 1 - \frac{\sum_{i=1}^{D} |TDF_Q(i) - TDF_S(i)|}{\sum_{i=1}^{D} \max(|TDF_Q(i)|, |TDF_S(i)|)} \tag{8}$$

$$SIM_{TA} = 1 - \frac{\sum_{i=1}^{D} |TAF_Q(i) - TAF_S(i)|}{\sum_{i=1}^{D} \max(|TAF_Q(i)|, |TAF_S(i)|)} \tag{9}$$

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

$$SIM_T = \alpha SIM_{TD} + (1 - \alpha) SIM_{TA} \tag{10}$$

where, α is a weighting factor ranging from 0 to 1. The final similarity between the query video clip and a suspicious clip is computed by

$$SIM_C = \beta SIM_S + (1 - \beta) SIM_T \tag{11}$$

where, the range of the weighting factor β is from 0 to 1. Then, by comparing the computed similarity SIM_C with a predefined threshold SIM_{TH}, 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 α and β 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* [37], 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.

$$\text{Recall} = \frac{N_{Positive}}{N_{Total}} \tag{12}$$

$$\text{Precision} = \frac{N_{Positive}}{N_{Detected}} \tag{13}$$

where $N_{Positive}$ represents the number of true samples detected, N_{Total} represents the total number of all samples (positive and negative samples) in the database, and $N_{Detected}$ 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.

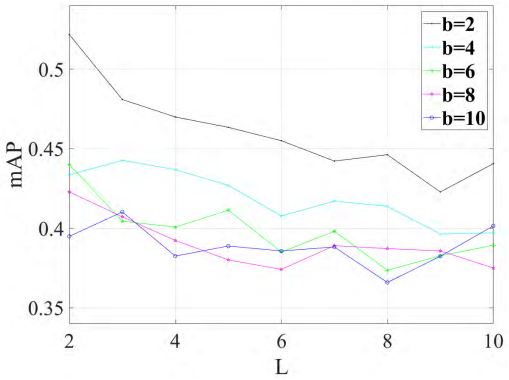


FIGURE 7. The effects of *L* and *b* on the mAP.

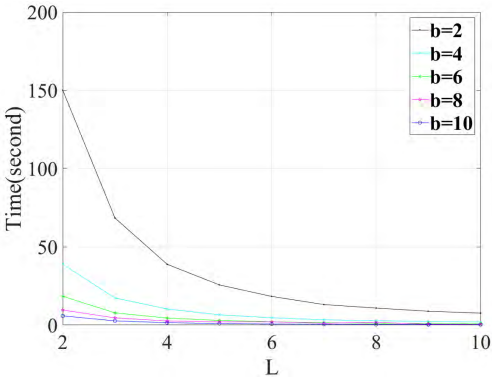


FIGURE 8. The effects of *L* and *b* on the time cost.

B. PARAMETER SELECTION

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

We first fix the parameters α and β 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 α and β by setting L and b to 7 and 10, respectively. The effects of the parameters α and β on the mAP is shown in Fig. 9. As shown in this figure, as the value of β 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.

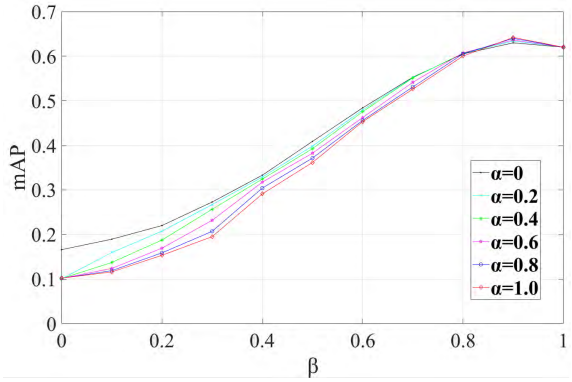


FIGURE 9. The effects of α and β on the mAP.

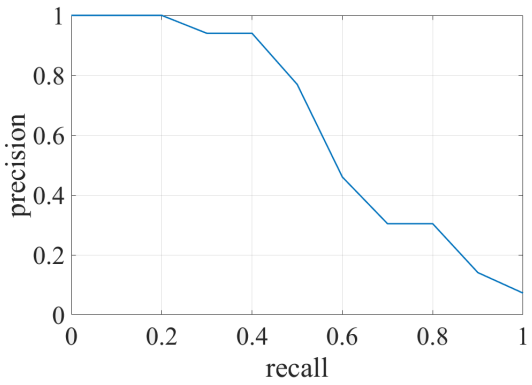


FIGURE 10. The PR curve of our copy detection method on the *TRECVID 2008* dataset.

TABLE 1. Comparison to the other methods.

	Ours	Sum-pooling	Max-pooling
mAP(%)	**0.653**	0.648	0.6431
Time(second)	0.87	0.56	0.58
Byte(MB)	6.63	6.27	6.27

Similarly, too large or too small α leads to inferior performance. According to the two figures, when the value of α is 0.9 and β is 0.9, the optimal mAP value is obtained. Thus, we set α and β 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 1. 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.

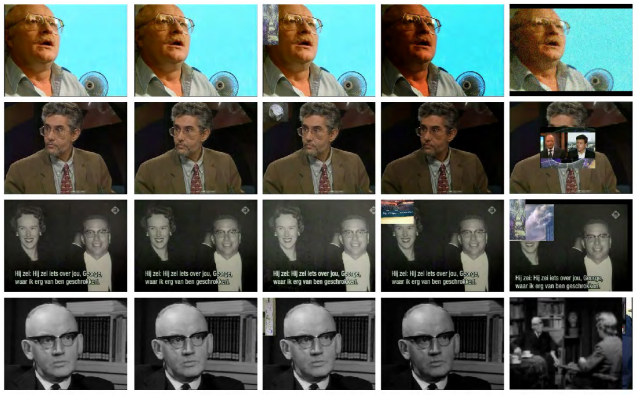


FIGURE 11. Some examples of detection results (four queries and the corresponding top ranked detected copies) using our method on the *TRECVID 2008* dataset. The examples demonstrate that our method can achieve desirable accuracy for copy detection.

In Table 1, 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 using the proposed spatio-temporal CNN features can achieve high performances in the aspects of both efficiency and effectiveness.

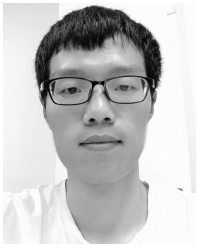
REFERENCES

[1] L. Qi, R. Wang, C. Hu, S. Li, Q. He, and X. Xu, "Time-aware distributed service recommendation with privacy-preservation," *Inf. Sci.*, vol. 480, pp. 354–364, Apr. 2018.
[2] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, pp. 1–23, May 2019. doi: 10.1007/s11280-019-00684-y.
[3] W. W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, Apr. 2018, Art. no. 3075849. doi: 10.1155/2018/3075849.
[4] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.
[5] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Gener. Comput. Syst.*, vol. 88, pp. 636–643, Nov. 2018.
[6] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *CMC, Comput., Mater. Continua*, vol. 55, no. 1, pp. 1–16, 2018.
[7] J. Cui, Y. Zhang, Z. Cai, A. Liu, and Y. Li, "Securing display path for security-sensitive applications on mobile devices," *CMC: Comput., Mater. Continua*, vol. 55, no. 1, pp. 17–35, 2018.
[8] Y. Liu, H. Peng, and J. Wang, "Verifiable diversity ranking search over encrypted outsourced data," *Comput., Mater. Continua*, vol. 55, no. 1, pp. 37–57, Jun. 2018.
[9] J. Lei, J. Duan, W. Feng, N. Ling, and C. Hou, "Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 706–718, Mar. 2018.
[10] J. Lei, B. Peng, C. Zhang, X. Mei, X. Cao, X. Fan, and X. Li, "Shape-preserving object depth control for stereoscopic images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3333–3344, Dec. 2018.
[11] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.
[12] Z. Pan, H. Qin, X. Yi, Y. Zheng, and A. Khan, "Low complexity versatile video coding for traffic surveillance system," *Int. J. Sensor Netw.*, vol. 30, no. 2, pp. 116–125, 2019.
[13] Z. Zhou, Q. M. J. Wu, and X. Sun, "Multiple distance-based coding: Toward scalable feature matching for large-scale Web image search," *IEEE Trans. Big Data*, to be published. doi: 10.1109/TBDATA.2019.2919570.
[14] Z. Zhou, Q. M. J. Wu, S. Wan, W. Sun, and X. Sun, "Intergrating SIFT and CNN feature matching for partial-duplicate image detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, to be published. doi: 10.1109/TETCI.2019.2909936.
[15] Z. Zhou, Y. Wang, Q. M. J. Wu, C.-N. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 48–63, Jan. 2017.
[16] Z. Zhou, C.-N. Yang, B. Chen, X. Sun, Q. Liu, and Q. M. J. Wu, "Effective and efficient image copy detection with resistance to arbitrary rotation," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 6, pp. 1531–1540, 2016.
[17] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 127–132, Jan. 2005.
[18] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 257–266, Jun. 2010.
[19] G. Özbulak, F. Kahraman, and S. Baykut, "Robust video copy detection in large-scale TV streams using local features and CFAR based threshold," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, Beijing, China, Oct. 2016, pp. 124–128.
[20] G. Yang, N. Chen, and Q. Jiang, "A robust hashing algorithm based on SURF for video copy detection," *Comput. Secur.*, vol. 31, no. 1, pp. 33–39, 2012.
[21] Y. Lei, W. Luo, Y. Wang, and J. Huang, "Video sequence matching based on the invariance of color correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1332–1343, Sep. 2012.
[22] M. Liu, L.-M. Po, C. Zhou, W. Y. F. Yuen, H.-K. Cheung, P. H. W. Wong, H.-T. Luk, and K.-W. Lau, "Content-based video copy detection using binary object fingerprints," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput.*, Qingdao, China, Sep. 2018, pp. 1–6.
[23] J. Li, H. Zhang, W. Wan, and J. Sun, "Two-class 3D-CNN classifiers combination for video copy detection," *Multimedia Tools Appl.*, pp. 1–13, May 2018. doi: 10.1007/s11042-018-6047-9.
[24] Y. Lou, Y. Bai, J. Lin, S. Wang, J. Chen, V. Chandrasekhar, L.-Y. Duan, T. Huang, A. C. Kot, and W. Gao, "Compact deep invariant descriptors for video retrieval," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Apr. 2017, pp. 420–429.
[25] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval by aggregating intermediate CNN layers," in *Proc. Int. Conf. Multimedia Modeling*, Cham, Switzerland, 2017, pp. 251–263.

[26] X. Zhang, Y. Xie, X. Luan, J. He, L. Zhang, and L. Wu, "Video copy detection based on deep CNN features and graph-based sequence matching," *Wireless Pers. Commun.*, vol. 103, no. 1, pp. 401–416, 2018.
[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[28] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jun./Jul. 2004, p. 2.
[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[31] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," in *Proc. Comput. Sci. Vis. Pattern Recognit.*, Oct. 2015, pp. 2911–2918.
[32] J. Y.-H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Boston, MA, USA, Jun. 2015, pp. 53–61.
[33] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Boston, MA, USA, Jun. 2015, pp. 36–45.
[34] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual instance retrieval with deep convolutional networks," *ITE Trans. Media Technol. Appl.*, vol. 4, no. 3, pp. 251–258, 2016.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[36] A. B. Yandex and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1269–1277.
[37] (2008). *TREC Video Retrieval Evaluation*. [Online]. Available: http://www-nlpir.nist.gov/projectsH/trecvid/

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 currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. He is also 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, coverless information hiding, digital forensics, and image processing.



JINGCHENG CHEN received the B.S. degree from the Binjiang College of Nanjing University of Information Science and Technology, China, in 2017. He is currently pursuing the M.S. degree with the Nanjing University of Information Science and Technology. His research interests include image retrieval, image/video copy detection, and information security.

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 the B.S. degree in mathematics from Hunan Normal University, China, in 1984, the M.S. degree in computing science from the Dalian University of Science and Technology, China, in 1988, and the Ph.D. degree in computing science from Fudan University, China, in 2001. In 2006, he visited the University College London, U.K.; he was a visiting Professor with the University of Warwick, U.K., from 2008 to 2010. He is currently a Professor with the College of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include network and information security, database security, and natural language processing.