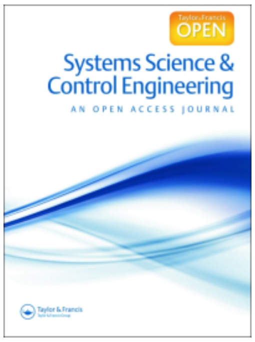


Taylor & Francis
Taylor & Francis Group

Systems Science & Control Engineering
An Open Access Journal

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tssc20

Video fingerprinting based on quadruplet convolutional neural network

Xinwei Li, Chen Guo, Yi Yang & Lianghao Xu

To cite this article: Xinwei Li, Chen Guo, Yi Yang & Lianghao Xu (2021) Video fingerprinting based on quadruplet convolutional neural network, Systems Science & Control Engineering, 9:sup1, 131-141, DOI: 10.1080/21642583.2020.1822946

To link to this article: https://doi.org/10.1080/21642583.2020.1822946

© 2020 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

Published online: 29 Sep 2020.

Submit your article to this journal

Article views: 1142

View related articles

View Crossmark data

Citing articles: 2 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tssc20

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL
2021, VOL. 9, NO. S1, 131–141
https://doi.org/10.1080/21642583.2020.1822946

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Video fingerprinting based on quadruplet convolutional neural network

Xinwei Li, Chen Guo, Yi Yang and Lianghao Xu

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, People's Republic of China

ABSTRACT

In order to achieve fast and accurate retrieval of video copies, this paper proposes a compact video fingerprinting based on quadruplet convolutional neural network. The algorithm consists of four branch networks with shared weights, each branch network contains feature extraction and quantization coding. The projection and excitation network is combined with 3D convolution for feature extraction, which mainly learns feature weights to improve useful features and suppresses valueless features. The deep features learned are mapped into approximate real vectors in a fully connected form and quantized to generate binary codes. The model employs an improved quadruplet loss to divide the feature distance between the copied videos and the non-copied, and a quantization error term is added to ensure that the fingerprint codes contains as much similar information as the original data. The experimental results performed on the public dataset show that the algorithm can effectively improve the robustness and distinctiveness, and the average detection accuracy under multiple compound attacks is better than the compared algorithms.

ARTICLE HISTORY
Received 28 April 2020
Accepted 8 September 2020

KEYWORDS
Video fingerprinting; 3D convolution; projection and excitation; quadruplet loss

1. Introduction

With the popularization of Internet and the development of computer technology, multimedia data have been conveniently spread. Taking videos as an example, while enriching human life and increasing human knowledge, some illegal content contained in it will infract the owner's copyright directly and affect the healthy development of society seriously (Gu et al., 2017). For this reason, various video copy detection technologies have been proposed successively. In order to reduce the computer memory and accelerate retrieval, video fingerprinting has gradually developed into an important part of video copy detection. Video fingerprint codes are obtained by extracting features from video and quantizing them into binary form, so as to represent a large amount of data with very little data (Oostveen et al., 2002). It usually requires satisfying robustness, distinctiveness and compactness. Robustness means that the extracted features are highly similar under some distortions; distinctiveness means that the codes are different from different videos; compactness is to enable the codes to be expressed as short as possible.

The key of video fingerprinting is how to effectively extract video features. Traditional video fingerprintings mainly rely on handcraft methods to extract features (De et al., 2005; Esmaeili & Ward, 2010; Lee & Yoo, 2006; Li & Monga, 2013; Malekesmaeili et al., 2009), there are problems such as complicated pre-processing and insufficient capture capabilities of spatio-temporal information. In recent years, as deep learning have made outstanding achievements in computer vision, researchers have tried to use convolutional neural network (CNN), long-short term memory (LSTM), recurrent neural network (RNN) and other neural networks to autonomously extract features, which has promoted the development of video fingerprinting. Jiang and Wang (Jiang & Wang, 2016) used pre-trained AlexNet to extract frame features, experimental results shown that it has superior performance than traditional methods. Wang, Bao and Li (Wang et al., 2017) used VGGNet to extract features and then reduced the dimensions through principle component analysis (PCA), and the performance was further improved. The above methodes are based on 2D CNN, which can only extract the spatial features but ignore the temporal information among consecutive frames. For this reason, Li and Chen (Li & Chen, 2017) trained conditional restricted boltzmann machine (CRBM) and denoising auto-encoder (DAE) respectively to obtain robust spatio-temporal descriptor. Li, Zhang and Wan (Li et al., 2018) used parallel 3D CNN to extract spatio-temporal features directly. Compared with 2D convolution, 3D convolution can capture motion information along the time

CONTACT Yi Yang chinayangyi@hpu.edu.cn; yangyi@hpu.edu.cn

© 2020 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

dimension, however, the common 3D CNN is difficult to mine deeper semantic information to satisfy compactness .

Due to the great memory consuming, some works try to combine deep learning and hashing to generate binary codes directly. Ma, Gu and Gong (Ma et al., 2018) used CNN and LSTM to extract video spatial and temporal features respectively, and fuses frame-level features into video-level features. Zhang, Wang and Hong (Zhang et al., 2016) integrated feature extraction and quantization coding under a unified framework, and proposed a binary LSTM unit. Compared with separated feature extraction and hash retrieval methods, the above end-to-end fingerprintings reduce the information loss in quantization. Guo, Li, Yang and Xu (Guo et al., 2019) proposed a video fingerprinting based on the triplet network, each subnetwork used a 3D residual network to simultaneously capture the global and local spatio-temporal information. The above algorithms have been shown to significantly improve performance under certain distortions, however, the triplet net is not enough to fully metric the similarity among videos.

Aiming at the deficiency of 3D convolution and triplet network, we builds quadruplet network to improve it. Specifically, each branch of quadruplet network uses 3D ResNet (Hara et al., 2017) with an implanted PE Block (Rickmann et al., 2019) as feature extractor. The PE Block can generate weights for feature channels. In addition, the output layer of the network is used to generate the fingerprint codes. The overall framework can realize the end-to-end mapping of original video to binary code. During training, both the improved quadruplet loss and the quantization error loss are jointly optimized.

The remainder of this paper is structured as follows. Section 2 shows the overall quadruplet framework. Section 3 describes the detailed quadruplet fingerprinting. Section 4 introduces the complete experimental process of model training and performance testing. Section 5 gives relevant conclusions and further work.

2. Framework design

As shown in Figure 1, the framework proposed in this paper is quadruplet network (Chen et al., 2017), where the input uses a set of quadruplet video, anchor represents the source video, positive corresponds to the copied video, negative1 and negative2 are non-copy videos. Each sub-network uses 3D ResNet-50 as the backbone, and project and excite block is implanted in the convolution layers. The high-dimensional features learned by the feedforward network are mapped to a real-valued vector with fixed length through a full connection, and then a binary code is obtained by the binarization process. The objective function composed of the improved

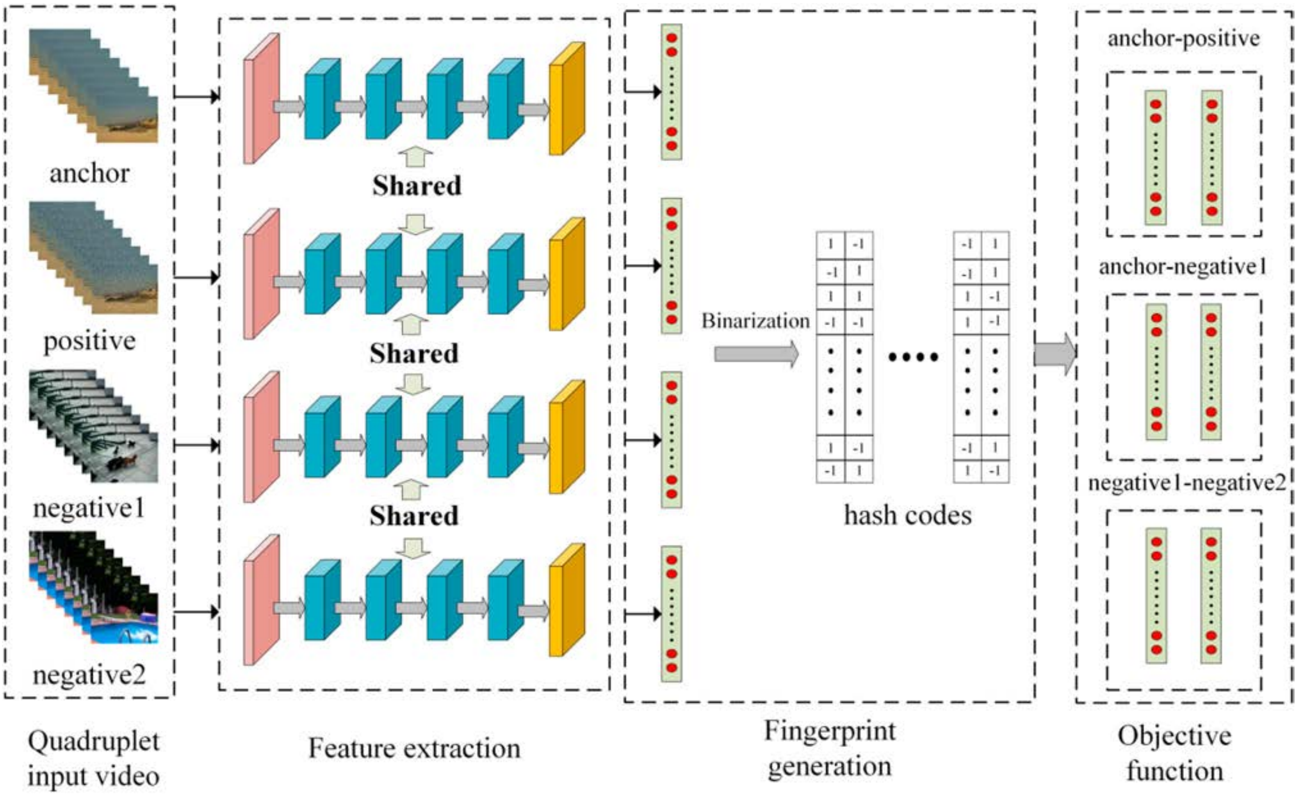


Figure 1. Framework of quadruplet fingerprinting.

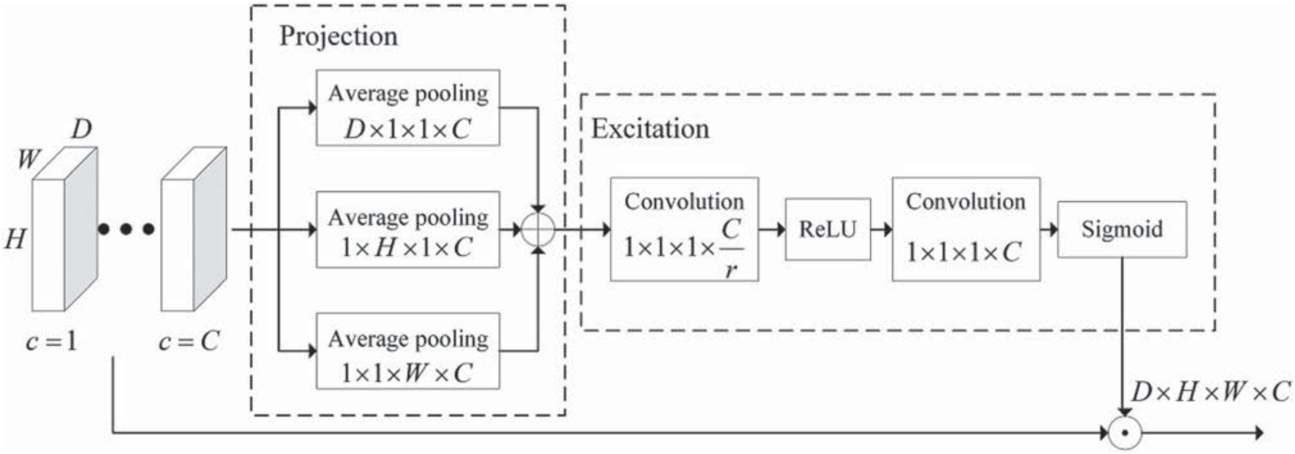

Projection
Average pooling $D\times1\times1\times C$
Average pooling $1\times H\times1\times C$
Average pooling $1\times1\times W\times C$
Excitation
Convolution $1\times1\times1\times\frac{C}{r}$
ReLU
Convolution $1\times1\times1\times C$
Sigmoid
D
W
H
$c=1$
$c=C$
$D\times H\times W\times C$

Figure 2. Projection and excitation network.

quadruplet loss and quantization error loss completes the model optimization to ensure that the fingerprint meets the robustness and distinctiveness.

2.1. Projection and excitation network

The feature maps obtained after each convolution in the convolutional neural network contain rich feature information. For traditional 3D CNNs(Tran et al., 2015), the information obtained in the current layer mainly comes from the feature fusion in the spatial and temporal dimensions of the previous feature maps , however, the difference information among the channels is often ignored. Therefore, this paper introduces projection and excitation network (PENet) (Rickmann et al., 2019) to achieve the feature fusion through channels . Specifically, each feature channel is assigned a weight, so that the model learns the relationship among the channels, so as to filter the channel features that meet the current task.

Figure 2 shows the principle of the projection and excitation network. The input is a feature map with $D \times H \times W \times C$. The calculation includes two steps. In the first step, the projection performs global average pooling along the three dimension directions D, H and W of each feature channel to obtain three projection vectors; $\oplus$ represents an addition operation, and it fuses spatial information in each direction. The second step to stimulate the excitation operation consists of two layers of $1 \times 1 \times 1$ convolutions; ReLU and Sigmoid functions are used to activate each convolution layer, and r is the dimensionality reduction coefficient. These operations can generate weights for each feature channel to grasp the importance. $\odot$ means dot multiplication operation.

2.2. End-to-end structure

According to the framework shown in Figure 1, each branch is an end-to-end structure that maps original video data to fingerprint codes. The structure and detailed parameters are shown in Figure 3, the Input is video data, the Conv1 layer uses a $7 \times 7 \times 7$ convolution kernel to obtain 64 feature maps, and the Max Pool layer uses a sliding window of size of $3 \times 3 \times 3$ for max pooling. Conv2_x, Conv3_x, Conv4_x and Conv5_x are stacked by 3, 4, 6, 3 repeated residual units respectively, each of which contains two $1 \times 1 \times 1$ and one $3 \times 3 \times 3$ convolution kernels. The PE Block is located in each residual unit, this embedding method does not change the basic network structure. More importantly, it can adaptively calibrate the feature channels, so that the network captures important channel feature information. Ave Pool layer uses a sliding window of size of $1 \times 4 \times 4$ for average pooling. The Output consists of two parts: the 2048-dimensional vector and the16-bit binary codee 1.

3. Algorithm description

3.1. Related definitions

The purpose of the video fingerprinting is to establish a mapping relationship, while keeping the code as compact as possible. Its formula is as follows:

$$H(v) = [h_1(v), h_2(v), \cdots, h_k(v)]^T \in \{-1, 1\}^k \quad (1)$$

where $h_i(\cdot)(i = 1, 2, \cdots, k)$ represents the mapping function, and $[\cdot]^T$ represents the transpose operation. In addition, we use $f(v;\Theta) \in R^{k\times1}$ to represent the k-dimensionvector extracted from the video v, where Θ is the model parameter; similarly, we use the sign function $\mathrm{sgn}(\cdot)$ to quantize and encode the real-value features.

3.2. New quadruplet loss

For the quadruplet framework, four videos v_a, v_p, v_{n1}, and v_{n2} form a quadruplet as the training sample, where v_a

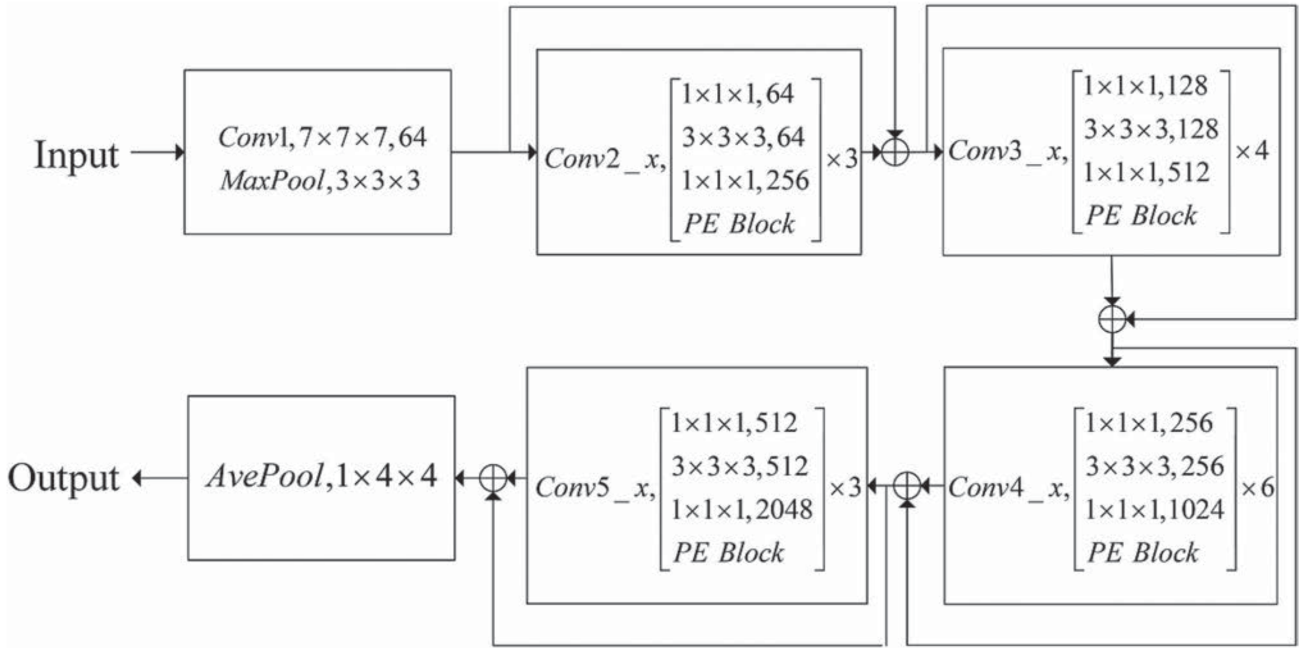


Figure 3. End-to-end structure and parameters.

and v_p are copied pair, v_a and v_{n1}, v_a and v_{n2} are the non-copy pairs. The quadruplet can extract the corresponding features $(f(v_a;\Theta), f(v_p;\Theta), f(v_{n1};\Theta), f(v_{n2};\Theta))$. This can further reduce the intra-class distance and increase the inter-class distance, prompting the model to generate higher quality video fingerprints. However, the traditional quadruplet loss (Chen et al., 2017) is optimized by the $\max(0,\cdot)$ function, which makes the function have undifferentiable points. In response to these problems, we design a new quadruplet loss function, that is, a smoother continuous function $\ln(1+\exp(\cdot))$ is used to replace the $\max(0,\cdot)$ for gradient calculation. The specific formula is:

$$\begin{aligned} L_1 = {} & \ln[1+\exp(f_e(v_a)-f_e(v_p)_2^2 - f_e(v_a)-f_e(v_{n1})_2^2 \\ & +\omega_1\alpha)] + \ln[1+\exp(f_e(v_a)-f_e(v_p)_2^2 - f_e(v_{n1}) \\ & - f_e(v_{n2})_2^2 + \omega_2\beta)] \end{aligned} \tag{2}$$

where $f_e(v_a), f_e(v_p), f_e(v_{n1})$, and $f_e(v_{n2})$ represent the normalized real-value features. The continuous feature vector is used as the optimization object of the quadruplet loss. It can simplify the calculation and avoid the undifferentiable situation when optimizing the discrete codes. In (2), $\|\cdot\|_2^2$represents the square Euclidean distance, $\omega_1\alpha$ and $\omega_2\beta$ represent the adaptive thresholds, and ω_1 and ω_2 are the corresponding thresholds. Using adaptive thresholds can better constrain the feature distance among various samples. α and β are determined by the quadruplet number during training. The mathematical expression is as follows:

$$\begin{aligned} \alpha &= \frac{1}{2N}\sum_{a,n1}^{N} f_e(v_a)-f_e(v_{n1})_2^2 - \frac{1}{N}\sum_{a,p}^{N} f_e(v_a)-f_e(v_p)_2^2\beta \\ &= \frac{1}{2N}\sum_{n1,n2}^{N} f_e(v_{n1})-f_e(v_{n2})_2^2 - \frac{1}{N}\sum_{a,p}^{N} f_e(v_a)-f_e(v_p)_2^2 \end{aligned} \tag{3}$$

where N is the training batch size. It calculates the average distance between the distribution of v_a and v_p, v_a and v_{n1} in the feature space to obtain α; Obtaining β is similar. The gradients of Equation (2) with respect to $f_e(v_a), f_e(v_p)$, $f_e(v_{n1})$ and $f_e(v_{n2})$ are:

$$\begin{aligned} \frac{\partial L_1}{\partial f_e(v_a)} = {} & \left[\left(2-\frac{2\omega_1}{N}\right)(f_e(v_a)-f_e(v_p)) \right. \\ & \left. + \left(-2+\frac{\omega_1}{N}\right)(f_e(v_a)-f_e(v_{n1}))\right] \\ & \times \frac{\exp(l_1)}{1+\exp(l_1)} + \left(2-\frac{2\omega_2}{N}\right)(f_e(v_a)-f_e(v_p)) \\ & \times \frac{\exp(l_2)}{1+\exp(l_2)} \end{aligned}$$

$$
\begin{aligned}
\frac{\partial L_1}{\partial f_e(v_p)} &= \left(-2+\frac{2\omega_1}{N}\right)(f_e(v_a)-f_e(v_p)) \times \frac{\exp(l_1)}{1+\exp(l_1)} \\
&\quad + \left(-2+\frac{2\omega_2}{N}\right)(f_e(v_a)-f_e(v_p)) \\
&\quad \times \frac{\exp(l_2)}{1+\exp(l_2)} \\
\frac{\partial L_1}{\partial f_e(v_{n1})} &= \left(2-\frac{\omega_1}{N}\right)(f_e(v_a)-f_e(v_{n1})) \times \frac{\exp(l_1)}{1+\exp(l_1)} \\
&\quad + \left(-2+\frac{\omega_2}{N}\right)(f_e(v_{n1})-f_e(v_{n2})) \\
&\quad \times \frac{\exp(l_2)}{1+\exp(l_2)} \\
\frac{\partial L_1}{\partial f_e(v_{n2})} &= \left(2-\frac{\omega_2}{N}\right)(f_e(v_{n1})-f_e(v_{n2})) \times \frac{\exp(l_2)}{1+\exp(l_2)}
\end{aligned} \tag{4}
$$

where

$$
\begin{aligned}
l_1 &= \| f_e(v_a)-f_e(v_p) \|_2^2 - \| f_e(v_a)-f_e(v_{n1}) \|_2^2 + \omega_1\alpha \\
l_2 &= \| f_e(v_a)-f_e(v_p) \|_2^2 - \| f_e(v_{n1})-f_e(v_{n2}) \|_2^2 + \omega_2\beta
\end{aligned} \tag{5}
$$

3.3. Quantization error loss

In the binarization process, in order to reduce the quantization error as much as possible, the following quantization error loss function based on quadruplet is used:

$$
\begin{aligned}
L_2 &= \| H(v_a)-f_e(v_a) \|_2^2 + \| H(v_p)-f_e(v_p) \|_2^2 \\
&\quad + \| H(v_{n1})-f(v_{n1}) \|_2^2 + \| H(v_{n2})-f(v_{n2}) \|_2^2
\end{aligned} \tag{6}
$$

where $H(v_a)$, $H(v_p)$, $H(v_{n1})$ and $H(v_{n2})$ represent the binary codes generated from $f_e(v_a)$, $f_e(v_p)$, $f_e(v_{n1})$ and $f_e(v_{n2})$, namely $H(v_a) = sign(f_e(v_a)), H(v_p) = sign(f_e(v_p))$, $H(v_{n1}) = sign(f_e(v_{n1}))$ and $H(v_{n2}) = sign(f_e(v_{n2}))$. The gradients of Equation (6) with respect to $f_e(v_a)$, $f_e(v_p)$, $f_e(v_{n1})$ and $f_e(v_{n2})$ are:

$$
\begin{aligned}
\frac{\partial L_2}{\partial f_e(v_a)} &= 2[f_e(v_a)-H(v_a)] \\
\frac{\partial L_2}{\partial f_e(v_p)} &= 2[f_e(v_p)-H(v_p)] \\
\frac{\partial L_2}{\partial f_e(v_{n1})} &= 2[f_e(v_{n1})-H(v_{n1})] \\
\frac{\partial L_2}{\partial f_e(v_{n2})} &= 2[f_e(v_{n2})-H(v_{n2})]
\end{aligned} \tag{7}
$$

3.4. Algorithm flow

The objective function integrates the above new quadruplet loss and quantization error loss. At the same time, in order to prevent overfitting during training, L1 regularization term is added to enhance the model generalization ability. Because L1 regularization term will impel many parameters to be close to 0, thus some important features are kept and the trivial are discarded. Then the common key features among the samples from training and testing set are intensified. The formula of the objective function is:

$$
\begin{aligned}
L &= \sum_{a,p,n1}^{N} \ln[1+\exp(f_e(v_a)-f_e(v_p)_2^2 - f_e(v_a) \\
&\quad - f_e(v_{n1})_2^2 + \omega_1\alpha)] \\
&\quad + \sum_{a,p,n1,n2}^{N} \ln[1+\exp(f_e(v_a)-f_e(v_p)_2^2 - f_e(v_{n1}) \\
&\quad - f_e(v_{n2})_2^2 + \omega_2\beta)] \\
&\quad + \lambda \sum_{a,p,n1,n2}^{N} (H(v_a)-f_e(v_a)_2^2 + H(v_p)-f_e(v_p)_2^2 \\
&\quad + H(v_{n1})-f(v_{n1})_2^2 + H(v_{n2})-f(v_{n2})_2^2) + \mu\Theta_1
\end{aligned} \tag{8}
$$

In (8), Θ_1 represents the sum of the absolute values of the parameters of each part of the model, and λ, μ are the weighted parameters.

The overall process of the PE_Quadruplet algorithm is as follows:

Input: Quadruplet video dataset $U = \{(v_a, v_p, v_{n1}, v_{n2})_i\}_{i=1}^{N}$.

Output: the updated model parameter Θ.

```
Initialization: the parameters $\Theta$ of each part of the model, mini-batch $=$ 10, iteration $=$ 90000, learning rate 0.01;
repeat
for iteration = 1,2, . . . ,90000 do
        randomly sample a mini-batch size training sample from dataset $U$;
        For each $(v_a, v_p, v_{n1}, v_{n2})$, $(f(v_a;\Theta), f(v_p;\Theta), f(v_{n1};\Theta), f(v_{n2};\Theta))$ is calculated by forward propagation and normalized to $(f_e(v_a), f_e(v_p), f_e(v_{n1}), f_e(v_{n2}))$;
        The sgn$(\cdot)$ calculates its corresponding code $(H(v_a), H(v_p), H(v_{n1}), H(v_{n2}))$;
        Calculate the corresponding gradient according to Equations (4) and (7);
        Update model parameter $\Theta$ through back propagation;
    end for
until the convergence condition is met.
```

4. Experiments

The experiment configuration includes Ubuntu 16.04, CPU of Inter core i7, 6 core, 3.70 GHz frequency and 32GB memory, Graphics card of RTX2080. In addition, the used deep learning framework is PyTorch.

4.1. Dataset

The video clips in this experiment are selected from three public datasets, UCF-101 (Soomro et al., 2012), HMDB-51 (Kuehne et al., 2011) and FCVID (Jiang et al., 2017).

The specific construction process of our dataset is as follows: firstly, three public datasets were selected with a resolution of 320 × 240, and a total of 4986 videos with large differences in visual; then, all the selected video clips are divided into training set and testing set, the number of training set and testing set are 3986 and 1000 respectively; finally, in order to ensure the convenience and uniformity, the first 100 frames of all selected video clips are intercepted, and the size of each sequence is normalized to 100 × 320 × 240.

4.2. Training

In order to successfully train the model, it is necessary to construct video quadruplet. It is to get three videos from the training set randomly, and select one to generate a distorted copy, and then form them to be training samples. According to the input need, each video is evenly sampled at equal intervals to obtain 16 frames. The four corners and the centre of the frame are respectively cropped at five scales $\left\{1, 1/2^{1/4}, 1/\sqrt{2}, 1/2^{3/4}, 1/2\right\}$ to obtain an image with size 112 × 112. During the experiments, the relevant parameters in the objective function are set as follows: the threshold coefficients ω_1 and ω_2 are 1 and 0.5, and the hyperparameters λ and μ are 0.01 and 0.001, respectively. It is should be noted that the selection of λ and μ will affect the results. because these two parameters are the weight values of quantization error term and L1 regularization term, so their changes will affect the optimization direction of the overall objective function. Through experiments, it is found that when λ and μ are 0.01 and 0.001 respectively, the convergence effect of the model is the best.

The training process adopts the following strategies to accelerate the convergence to achieve the best effect: (1) The PE Block parameters are randomly initialized. In addition, we use pre-trained 3D ResNet-50 parameters by the Kinetics dataset (Carreira & Zisserman, 2017) for initial assignment; (2) Model training uses stochastic gradient descent (SGD), where the momentum is set to 0.9 and the weight attenuation is set to 0.001; (3) According to the computer configuration, each epoch selects 10,000 quadruplets; (4) The learning rate is adjusted according to the iterations. When iterations reaches 20,000, the learning rate drops to 0.1 times of the original one.

We compare the complete training process of the PE_Quadruplet algorithm with the NL_Triplet algorithm (Guo et al., 2019) to intuitively reflect the entire training. To be fair, the NL_Triplet algorithm is trained on the same dataset and iterations. Figures 4 and 5 show the changing curves of loss and accuracy with increasing iterations. It can be seen from Figure 4, both networks can fit the video data very well. It can be more clearly observed through Figure 5 that the PE_Quadruplet recognition accuracy is significantly higher than that of NL_Triplet.

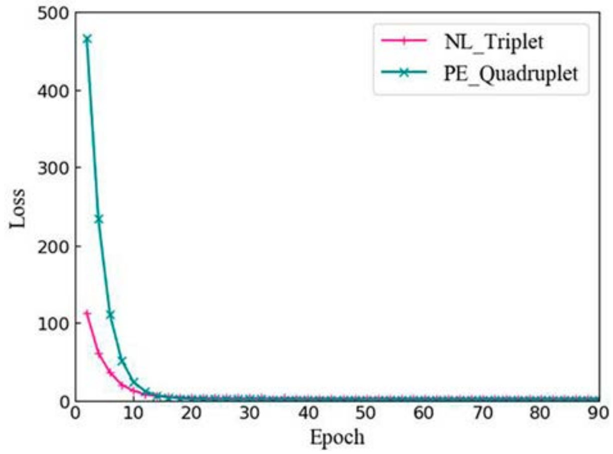


Figure 4. Train loss.

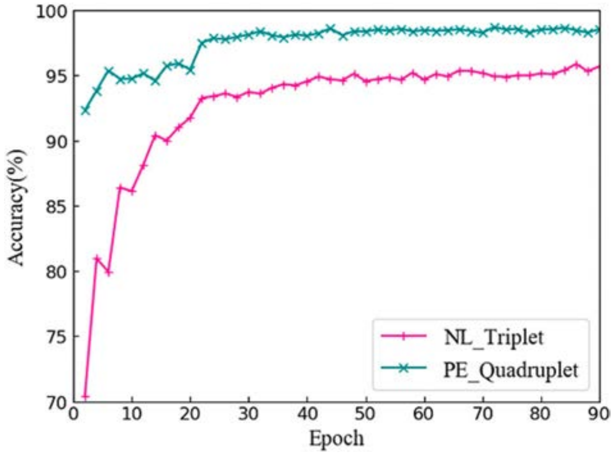


Figure 5. Train accuracy.

4.3. Testing

The test experiment is mainly to evaluate the detection accuracy under various distortions. Specifically, one or two simulated distortions are performed on each video in the testing set to generate 8 copies including single and combined types. The distortions include three aspects: spatial distortion, geometric distortion, and temporal distortion. The specific distortions and parameters are shown in Table 1.

For the evaluation index of the algorithm performance, we choose the receiver operator characteristic (ROC)

Table 1. Distortions and parameters.

Distortions	Parameters
Frame drop	Number of frames randomly dropped: 30, 40, 50
FPS reduction	Number of reduction per second: 2, 4, 6, 8, 10
Rotation	Clockwise: 5, 10 deg, Counterclockwise: 5, 10 deg
Shift plus FPS reduction	Move along the coordinate axis: (−40, 40), (−60, 60), (−80, 80), and number of reduction per second: 10
Insert logo	Insert position coordinates: (40, 60), (60, 80), (80, 100), (100, 120), (120, 140)
Median blur plus Frame drop	Filter size: 9×9, 11×11, 13×13, 15×15, 17×17, and number of frames randomly dropped: 50
Salt and pepper noise plus Frame drop	Density: 0.02, 0.04, 0.06, 0.08, 0.10, and number of frames randomly dropped: 50
Scaling	Size: 0.5, 0.75, 1.5, 2.0

curve and F_1 score. The specific calculation formula is as follows:

$$F_1 = \max\left\{\frac{2 \times [1 - FPR(\eta)] \times TPR(\eta)}{1 - FPR(\eta) + TPR(\eta)}\right\},\quad \eta \in [d_{\min}, d_{\max}] \tag{9}$$

where max{·} means taking the maximum value, and η means 100 thresholds selected at equal intervals between $d_{\min}$ and $d_{\max}$ in the Hamming distance. The higher the F_1 score calculated using Equation (9), the better the algorithm performance, and vice versa.

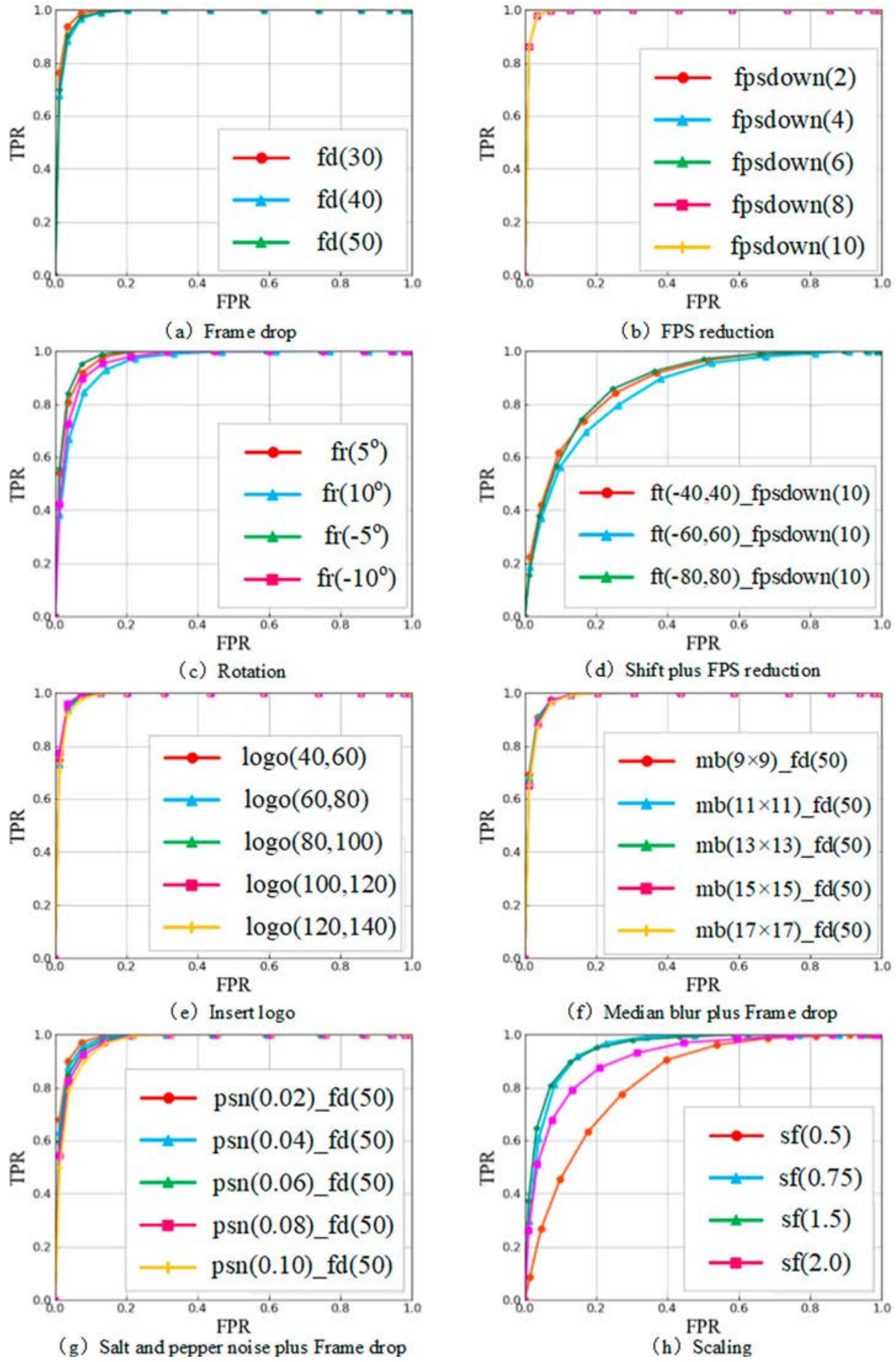
In order to comprehensively verify the performance of the algorithm, we conducted two groups of experiments: the first group is the comparison of our algorithm under kinds of distortion; the second group is the comparison among several algorithms.

Figure 6 shows the ROC curves for distortions. Among them, Figure 6(a) is the graph of frame dropping. It can be seen that the performance decreases slightly with the increasing of frame dropping. Figure 6(b) is the graph of FPS reduction. Obviously, the change of the FPS has little effect on the performance, indicating that the combination of the 3D residual network and PE Block can better capture the frame correlation. Figure 6(c) is the case of frame rotation. It can be found that the larger the rotation angle, the worse the algorithm performance. Figure 6(d) is the graph of video frame shifting plus FPS reduction. In the initial stage, as the frame shifting position moves away, the performance shows the downtrend. However, when the translation position reaches a certain distance the performance rises, indicating that the combination of shifting and FPS reduction does not necessarily reduce the performance. Figure 6(e) is the graph when a logo is inserted. The closer the inserted logo is to the centre of the video frame, the worse the anti-interference ability of the algorithm. Conversely, the closer the logo position is to the edge of the video frame, the more anti-interference is strong, this is because most of the key information contained in the centre area of the video frame. Figure 6(f) and (g) corresponds to the graphs of median blur plus frame dropping, salt &pepper noise plus frame dropping, respectively.

It can be seen that no matter whether it is median blur or salt&pepper noise, when the intensity is improved, the performance does not decrease sharply. This is because the 3D residual network has an outstanding learning ability for the pixel-level features. Even if a part of the video frames are additionally discarded, the robustness and distinctiveness of the algorithm still perform well under the distortions of these two types. Figure 6(h) is the graph of the video frame scaling. It can be seen that scaling the video frame according to a certain factor will obviously affect the performance, It means that shrinking the frame loses more spatial structure information than expanding it.

The second group of experiments is to compare our PE_Quadruplet algorithm with four classic video fingerprintings, RASH (De et al., 2005), CGO (Lee & Yoo, 2006), TIRI (Esmaeili & Ward, 2010) and SGM (Li & Monga, 2013), and the NL_Triplet algorithm (Guo et al., 2019) based on deep learning. The NL_Triplet uses the same dataset as the PE_Quadruplet for experiments, and the codes length for them is set to 16 bits. Figure 7 shows the ROC curves of these algorithms. It can be seen that the overall detection performance of our algorithm has improved compared with others . Specifically, in the frame dropping case shown in Figure 7(a), our algorithm is significantly better than the TIRI, CGO and RASH, which shows that the 3D residual network combined with PE Block can still better grasp the linkage among frames when the temporal information is damaged. As shown in Figure 7(b), when the FPS reduces, our algorithm is slightly inferior to the SGM and NL_Triplet, but the overall performance is still at a high level. This is because the 3D convolutional network and the projection and excitation network have limited levels of information acquisition in the temporal and channel dimensions, resulting in the recognition effect when the FPS changes are not particularly significant.

Figure 7(c) shows that our algorithm is the best compared in the rotation situation, because the proposed quadruplet loss is used to optimize the model to further improve the resistance ability to geometric attacks. Figure 7(d) shows that under the double attacks of frame shifting and FPS reduction, our performance is significantly higher than that of other algorithms. It can be seen that the PE_Quadruplet algorithm is more prominent in the anti-interference ability of geometric and time-domain combined distortion. Figure 7(e) shows that in the case of inserting a logo, the PE_Quadruplet algorithm is still the best among all algorithms, indicating that it is also very robust against local distortion. Figure 7(f) shows that



(a) Frame drop

(b) FPS reduction

(c) Rotation

(d) Shift plus FPS reduction

(e) Insert logo

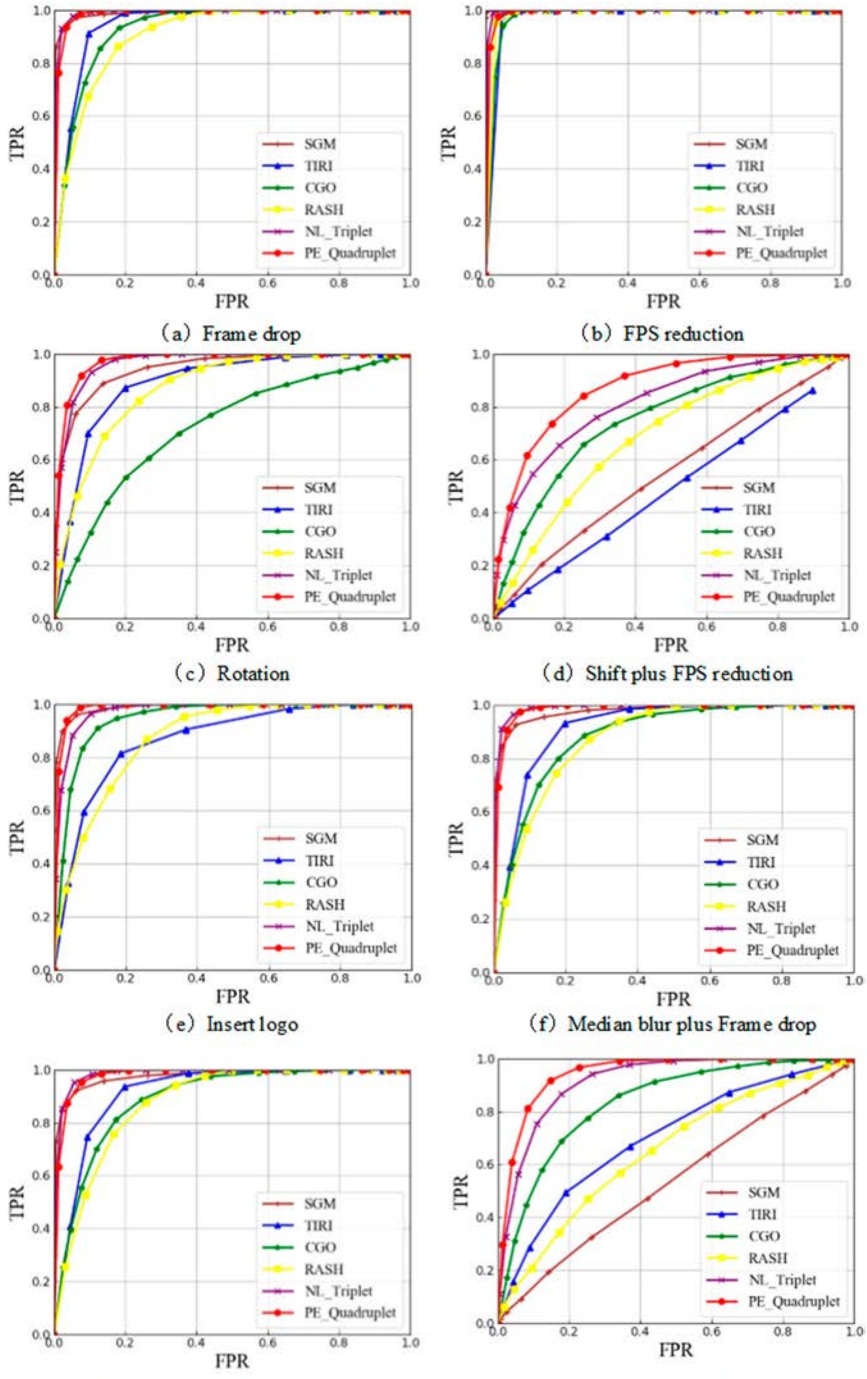
(f) Median blur plus Frame drop

(g) Salt and pepper noise plus Frame drop

(h) Scaling

Figure 6. The ROC for all distortions.

under the double attacks of median blurring plus frame dropping, our algorithm exceeds four traditional algorithms, and it is almost the same as the deep learning algorithm NL_Triplet, indicating that the PE_Quadruplet algorithm is excellent in the robustness of spatial distortion of blur-type. Figure 7(g) shows that under the double attacks of salt &pepper noise plus frame dropping, although our performance is slightly inferior to the deep learning algorithm NL_Triplet, but it is slightly better than the traditional algorithms. It reflects that the PE_Quadruplet algorithm has certain robustness to noise-type spatial distortion. As shown in Figure 7(h), in the scaling situation, the algorithm has the best detection effect compared with the other algorithms, which shows the strong robustness and high distinctiveness against geometric distortion.



(a) Frame drop

(b) FPS reduction

(c) Rotation

(d) Shift plus FPS reduction

(e) Insert logo

(f) Median blur plus Frame drop

(g) Salt and pepper noise plus Frame drop

(h) Scaling

Figure 7. The ROC for contrast algorithms with different distortions.

In addition, Table 2 shows the F_1 scores of several algorithms. Obviously, compared with other algorithms, our F_1 scores are the best. This shows that the fingerprint-codes generated by the quadruplet network are compact, at the same time, it has outstanding robustness and distinctiveness, which enables fast and accurate retrieval of video copies.

5. Conclusion

The algorithm in this paper combines deep learning and hashing, and uses the 3D ResNet with PE Block embedded to learn the semantic similarity features of the quadruplet video. In order to facilitate the calculation of gradients, the designed new quadruplet loss and quantization error loss are used to jointly train the model.

Table 2. F_1 scores comparison of several algorithms.

Distortions	PE_Quadruplet	NLTriplet	SGM	TIRI	CGO	RASH
Frame drop	0.9557	0.9613	0.9550	0.9083	0.8712	0.8413
FPS reduction	0.9712	0.9856	0.9972	0.9672	0.9501	0.9627
Rotation	0.9215	0.9126	0.8753	0.8351	0.6736	0.7914
Shift + FPS reduction	0.7909	0.7337	0.5334	0.4917	0.6988	0.6432
Insert logo	0.9564	0.9303	0.9484	0.8143	0.8938	0.7996
Median + Frame drop	0.9498	0.9550	0.9321	0.8626	0.8107	0.7966
Median + Frame drop	0.9377	0.9479	0.9298	0.8644	0.8189	0.8037
Scaling	0.8843	0.8437	0.5205	0.6487	0.7617	0.6100

When experimentally verifying the feasibility of the algorithm, it is found that the overall effect of the quadruplet training method has indeed improved, however, its performance is not very satisfactory in terms of spatial domain distortion of signal processing such as adding noise and blurring, indicating the end-to-end network still cannot achieve the expected effect for individual distortions. Therefore, the research will focus on the video autoencoder to extract better spatial–temporal features.

Acknowledgements

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China (No. 61402152), Programs for Science and Technology of Henan Province (No. 192102210099, No. 192102210100), National Key Research and Development Project (No. 2018YFC0604502).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China [grant number 61402152], Programs for Science and Technology of Henan Province [grant numbers 192102210099, 192102210100], National Key Research and Development Project [grant number 2018YFC0604502].

References

Carreira, J., & Zisserman, A. (2017). *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*. Proceedings of the IEEE Conference on Computer Vision and Pattern RecognitionJuly 21-26.

Chen, W., Chen, X., & Zhang, J. (2017). *Beyond triplet loss: A deep quadruplet network for person re-identification*. Proceedings of the IEEE Conference on Computer Vision and Pattern RecognitionJuly 21-26.

De, R., De, V., & Lefebvre, F. (2005). Robust video hashing based on radial projections of key frames. *IEEE Transactions on Signal Processing* , *53*(10), 4020–4037. ‘10.1109/TSP.2005.855414

Esmaeili, M., & Ward, R. (2010). *Robust video hashing based on temporally informative representative images*. International Conference on Consumer Electronics, IEEE, January 9-13.

Gu, J., Zhao, R., & Jiang, Y. (2017). Video copy detection method: A review. *Journal of Computer Research and Development*, *54*(6), 1238–1250. 10.7544/issn1000-1239.2017.20170003.

Guo, C., Li, X., Yang, Y., & Xu, L. (2019). Video fingerprinting algorithm based on non-local 3D residual network. *Computer Engineering and Applications*, 1–12.

Hara, K., Kataoka, H., & Satoh, Y. (2017). *Learning spatio-temporal features with 3D residual networks for action recognition*. Proceedings of the IEEE International Conference on Computer Vision WorkshopsOctober 22-29.

Jiang, Y., & Wang, J. (2016). Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE Transactions on Big Data*, *2*(1), 32–42. ‘10.1109/TBDATA.2016.2530714

Jiang, Y., Wu, Z., & Wang, J. (2017). Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(2), 352–364. ‘10.1109/TPAMI.2017.2670560

Kuehne, H., Jhuang, H., & Garrote, E. (2011). *HMDB: A large video database for human motion recognition*. International Conference on Computer Vision, IEEE, November 1.

Lee, S., & Yoo, C. (2006). *Video fingerprinting based on centroids of gradient orientations*. IEEE International Conference on Acoustics Speech and Signal Processing ProceedingsMay 14-19. 2, II-II.

Li, Y., & Chen, X. (2017). *Robust and compact video descriptor learned by deep neural network*. International Conference on Acoustics, Speech and Signal Processing, IEEE, June 19.

Li, M., & Monga, V. (2013). Compact video fingerprinting via structural graphical models. *IEEE Transactions on Information Forensics and Security*, *8*(11), 1709–1721. ‘10.1109/TIFS.2013.2278100

Li, J., Zhang, H., & Wan, W. (2018). Two-class 3D-CNN classifiers combination for video copy detection. *Multimedia Tools and Applications*, *5*(4), 1–13. 10.1007/s11042-018-6047-9.

Ma, C., Gu, Y., & Gong, C. (2018). Unsupervised video hashing via deep neural network. *Neural Processing Letters*, *47*(3), 877–890. ‘10.1007/s11063-018-9812-x

Malekesmaeili, M., Fatourechi, M., & Ward, R. (2009). *Video copy detection using temporally informative representative images*. International Conference on Machine Learning and Applications, IEEE, December 13-15.

Oostveen, J., Kalker, T., & Haitsma, J. (2002). *Feature extraction and a database strategy for video fingerprinting*. International Conference on Advances in Visual Information Systems, Berlin, Springer, March 11-13.

Rickmann, A., Roy, A., & Sarasua, I. (2019). *'Project & Excite' modules for segmentation of volumetric medical scans*. International Conference on Medical image Computing and Computer-Assisted Interventio*n* (pp. 39–47). Cham, Springer.

Soomro, K., Zamir, A., & Shah, M. (2012). UCF101: a dataset of 101 human actions classes from videos in the wild. *Computer Science*, *3*(12), 1–9.

Tran, D., Bourdev, L., & Fergus, R. (2015). *Learning spatiotemporal features with 3d convolutional networks*. Proceedings of the IEEE International Conference on Computer VisionDecember 7-13.

Wang, L., Bao, Y., & Li, H. (2017). *Compact CNN based video representation for efficient video copy detection*. International Conference on Multimedia Modeling (pp. 576–587). Cham, Springer.

Zhang, H., Wang, M., & Hong, R. (2016). *Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing*. Proceedings of the 24th ACM International Conference on MultimediaOctober 1.