2021 International Conference on Culture-oriented Science & Technology (ICCST) | 978-1-6654-4254-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCST53801.2021.00117

Video Retrieval Based on CNN Feature and Scalar Quantization

Junlin Che
Communication University of China
Beijing,China
13512216092@163.com

Guixuan Zhang
Institute of Automation, Chinese Academy of Sciences
Beijing,China
guixuan.zhang@ia.ac.cn

Shuwu Zhang
Institute of Automation, Chinese Academy of Sciences
Beijing,China
shuwu.zhang@ia.ac.cn

Abstract—In recent years, the video dissemination has become an important information medium with the development of the Internet and the rise of short video platforms, and infringements against long videos have followed, so an method of efficient and automated short video infringement detection is necessary. This paper proposes a method of video copyright detection based on CNN features and Scalar Quantizer, in which the deep convolutional neural network is used to obtain the decoded video frame's feature vector, and then the Scalar Quantizer is used to search the feature vector based on the approximate nearest neighbor search, and finally the target video is determined by finding the shortest average Euclidean distance of the target video frames. This paper sets a distance threshold and a ratio threshold based on this method to form a new method, and then compares the recall and precision of two methods.

Keywords-Video Retrieval; Video Copyright Monitoring; CNN Feature; Scalar Quantizer

I. Introduction

In recent years, with the popularization of mobile terminals and the rapid development of networks, short videos have increasingly gained the favor of platforms and fans. At the same time, unauthorized editing, dissemination of original works and other infringements have emerged one after another in the short video industry. Video retrieval as a technology of the video feature extraction and the video feature similarity measurement has a good performance in the retrieval of infringing short videos. Due to the diversification and large-scale of infringements, infringing videos often undergo complex transformations. The corresponding video retrieval algorithm model must be qualified with a certain degree of robustness, a relatively fast execution speed and concurrency. Video retrieval refers to inputting a video segment and finding one or more videos which are similar to the input video in the database and returning them to the user. Video retrieval is divided into the text-based video retrieval and the content-based video retrieval, and the content-based video retrieval is generally divided into video data structure analysis, video feature extraction, and retrieval. The development of video retrieval systems at home and abroad has gradually matured. Among them, the classic video retrieval systems include IBM's QBIC image and video retrieval system[1], the Visual SEEK video retrieval system developed by Columbia University, and the TV-FI video program management system developed by Tsinghua University.

This paper mainly completes the retrieval of infringing short videos by extracting video features and searching the feature vector by the approximate nearest neighbor approach to match feature vectors. And the extraction of video features includes decoding, frame extraction, and feature vector calculation. This paper uses the convolutional layer features of the deep convolutional neural network and the feature extraction of Gaussian_Mac method to increase the weight of the central part of the image, which can extract the effective feature vector better, and then uses Scalar Quantizer to search the feature vector, and finally the target video is determined by finding the shortest average Euclidean distance of the target video frames. This paper sets a distance threshold and a ratio threshold based on this method to form a new method, and then compares the recall and precision of two methods.

II. Prerequisite Knowledge

A. Convolutional Neural Network

Convolutional neural network is a kind of feedforward neural network that includes convolutional calculation and has a deep structure. The basic structure of convolutional neural network includes input layer, convolution layer, excitation layer, pooling layer and fully connected layer. Compared with traditional neural networks, CNN reduces the number of parameters and complexity of the model through operations such as convolution, pooling, and nonlinear mapping, which improves the ability of the network to achieve fast backpropagation and the performance of parameter tuning. Therefore, a batch of applicable convolutional neural networks with excellent performance have emerged in the fields of image

978-1-6654-4254-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCST53801.2021.00117

classification, target recognition, image retrieval, and video retrieval, such as AlexNet, VGG-Net, GoogleNet, and ResNet.

The convolutional layer of the convolutional neural network is to extract features from different regions of the input data, and the extraction of high-level feature information is obtained as the convolutional layer deepens[2].

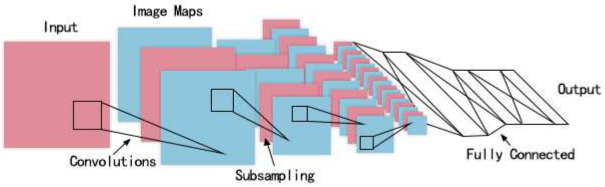


Fig.1 Basic frame diagram of convolutional neural network

Two-dimensional data is usually used as input. The relationship between feature matrix size F and input data size I, convolution kernel size C, and step size S is:

$$F = \frac{I - C}{S} + 1 \tag{1}$$

The input data and the data corresponding to the overlapping area of the convolutional kernel are calculated to obtain the output data. At the same time, a sliding operation with a step length of S is performed to obtain the next round of overlapping area calculation output, and finally a feature matrix is formed in this cyclic way. The pooling layer of the convolutional neural network is the next layer of the convolutional layer. The pooling function performs information filtering and feature selection on the feature data of the previous layer which plays the role of data compression and important feature extraction. Common pooling methods include Max-pooling, Mean-pooling, Stochastic-Pooling, etc.

B. Vector Index

The vector index is also called the approximate nearest neighbor search, which refers to constructing a data index structure of vectors that is efficient in time and space through a certain mathematical quantification model, so that we can obtain the K vectors that are closest to the query target vector in real time[3]. To obtain an efficient vector index model, three basic conditions are generally satisfied:

1) Real-time query, which can support real-time query of tens of billions or hundreds of billions of data.

2) Efficient storage, that is, the constructed vector index model has a high data compression ratio, achieving the goal of greatly reducing memory usage.

3) Highly recall, that is, compared with the brute force search, its top@K has a better recall accuracy.

The approximate nearest neighbor search is to divide the whole space into multiple subspaces, then quickly lock the target subspace through a certain search method, and finally traverse in the target subspace, while the brute force search only searches in the whole space. Obviously, compared with the brute force search, it can reduce the traversal space range, process large-scale data more effectively, and improve the search rate[4]. The basic vector index methods can be divided into four categories, namely the tree-based method, the hash method, the vector quantization method, and the graph index quantization method.

III. Algorithm Design

Since infringing short videos are mostly obtained by intercepting partial fragments of the original video and then performing composite transformation, the core task of searching for infringing short videos is to find video fragments with the same content and similar images between the infringing short video and the original video. The same content means that the viewers get the same semantic information when watching two videos; the similarity of the image means different forms of image deformation, such as resolution conversion, screen clipping, saturation changing, subtitles and logo addition, etc. This paper uses the proposed algorithm to find the recall and the precision of the original video corresponding to the infringing short video to measure the pros and cons of the algorithm. This chapter mainly introduces the specific steps of the algorithm. The specific experimental part will be elaborated in the next chapter.

A. Feature Extraction

For the video feature extraction, it usually includes three steps: decoding, frame extraction, and calculation of feature vectors[5]. Among them, decoding is to obtain image frames from videos; frame extraction is the process of retaining key frames and discarding non-key frames; the calculation of feature vectors is obtained according to the corresponding feature extraction method. This paper uses a deep convolutional neural network to extract the semantic features of image frames. Since this algorithm is suitable for a variety of CNN feature extraction, here is used to extract the last layer's convolutional features of Resnet18, and then using the Gaussian- Mac[6] method to aggregate the C*W*H dimensional features to obtain normalized 2C dimensional features. The specific steps are:

1) The generation of Gaussian kernel function, the formula is as follows:

$$\alpha(x, y) = \exp\left(- \frac{\left(y - \frac{H}{2}\right)^2 + \left(x - \frac{W}{2}\right)^2}{2\sigma^2} \right) \tag{2}$$

2) The Gaussian weight of each channel must stay the same when performing feature extraction on the convolutional layer, that is:

$$f(x, y, i) \times \alpha(x, y), 1 \le i \le C \tag{3}$$

3) Using the Maximum activations of convolutions method to aggregate data and reduce the dimension of feature vectors to achieve fast matching.

B. Searching Method

This paper adopts the Scalar Quantizer which is based on the approximate nearest neighbor search method. This method quantizes and encodes the dimensions of the original vector, which makes the error between the quantized vector and the original vector is decreased, the vector storage space is reduced, the read performance of the vector is acclerated, and the the vector index rate is also expedited. Scalar Quantization is mainly divided into three processes: training process, encoding process and decoding process[7]. The training process refers to train the parameters which is required for the encoding process,that is, the maximum and minimum values corresponding to each dimension; the encoding process refers to quantizing the float-type vectors into the int8-type vectors (taking the int8-type data compression form as an example, also it can be int4 type, etc.); the decoding process refers to decoding the encoded int8 type to the original float type. The Scalar Quantization diagram is as follows:

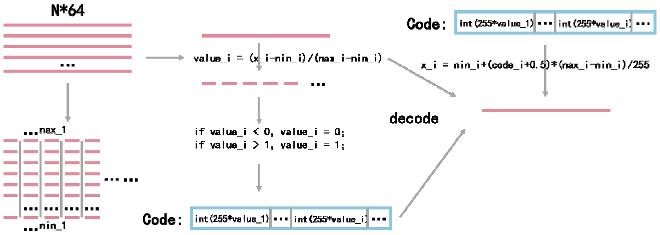

N*64
...
value_i = (x_i-nin_i)/(nax_i-nin_i)
...
...nax_1
if value_i < 0, value_i = 0;
if value_i > 1, value_i = 1;
......
...nin_1
Code: int(255*value_1) ... int(255*value_i) ...
decode
Code: int(255*value_1) ... int(255*value_i) ...
x_i = nin_i+(code_i+0.5)*(nax_i-nin_i)/255

Fig.2 The Scalar Quantization diagram

The specific steps are explained in reference to the schematic diagram are:

1) Training

The training process is mainly to obtain the maximum and minimum values of each dimension from samples, and then save them for the next stage of coding.

2) Encoding

In the process of encoding, taking a d-dimensional float-type vector $x = \left\{x\_1, x\_2, \ldots, x\_d\right\}$ which is encoded an int8-type vector as an example. The first thing is to find the each dimension of data, that is:

$$value\_i = \left(x\_i - \min\_i\right) \Big/ \left(\max\_i - \min\_i\right) \tag{4}$$

If $value\_i < 0$, then setting $value\_i = 0$; if $value\_i > 1$, then setting $value\_i = 1$ .Then encoding each dimension of data, that is:

$$code\_i = \text{int}\left(255 * value\_i\right) \tag{5}$$

In this way, each dimension of the float vector is encoded as an integer vector of int8 type.

3) Decoding

The decoding process of scalar quantization is the inverse process of encoding, that is:

$$x\_i = \min\_i + \left(code\_i + 0.5\right) * \left(\max\_i - \min\_i\right) \Big/ 255 \tag{6}$$

C. Two Matching Algorithm

For the convenience of description, the infringing short video is now referred to as the query, and the infringed video is referred to as the source. To complete the retrieval between videos, the idea of this paper is to implement similarity measurement on the basis of the image level to complete the nearest neighbor matching. In order to have a better comparison of matching results, this paper proposes two matching algorithms.

1) The algorithm 1

Step1:Performing different levels of frame extraction processing on the query and the source, and then performing feature extraction on the extracted key frames to calculate the corresponding feature vector. Supposing there are M key frames of the query video, and N source videos.

Step2: Using Scalar Quantizer to search a query video among the source video library. Since the unique nearest neighbor search is set, each query frame of the query video has the closest Euclidean distance to the key frame of the corresponding source video. According to the search results, one source video will correspond to M Euclidean distances, namely $X = \left\{D_1, D_2, \ldots D_M\right\}$.

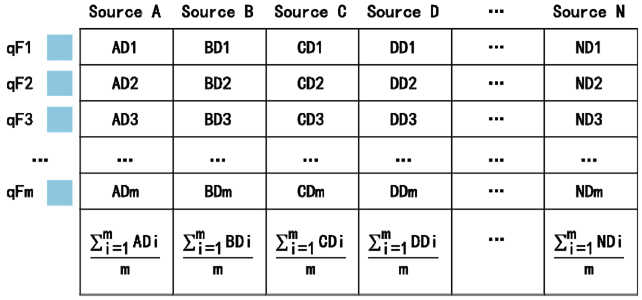
Step3: The M Euclidean distances obtained from a source video corresponding to the query video are averaged at the image frame level, namely:

$$\overline{D} = \frac{D_1 + D_2 + \cdots + D_M}{M} \tag{7}$$

Step4: Since a query video needs to be searched for N cycles in the source video library, N average Euclidean distances will be obtained, and a matching distance threshold $D_{\max}$ is set to filter out the nearest average Euclidean distance under the condition of meeting the threshold. If the purpose is to obtain the actual retrieval object, the index value corresponding to the average Euclidean distance is the source video t_source retrieved from the video to be queried, namely:

$$source = \underset{i}{\arg\min}\ \overline{D_i}, i \in [1, N] \tag{8}$$

The algorithm 1 diagram is as follows:



		Source A	Source B	Source C	Source D	···	Source N
qF1	■	AD1	BD1	CD1	DD1	···	ND1
qF2	■	AD2	BD2	CD2	DD2	···	ND2
qF3	■	AD3	BD3	CD3	DD3	···	ND3
···		···	···	···	···	···	···
qFm	■	ADm	BDm	CDm	DDm	···	NDm
		$\frac{\sum_{i=1}^{m} ADi}{m}$	$\frac{\sum_{i=1}^{m} BDi}{m}$	$\frac{\sum_{i=1}^{m} CDi}{m}$	$\frac{\sum_{i=1}^{m} DDi}{m}$	···	$\frac{\sum_{i=1}^{m} NDi}{m}$

Fig.3 The algorithm 1 diagram

2) The algorithm 2

In order to make the algorithm comparative and to select the algorithm suitable for this experiment, this paper proposed another matching algorithm which is modified on the basis of the algorithm 1. The specific steps are:

Step1: Performing different levels of frame extraction on the query and the source, and then performing feature extraction on the extracted key frames to calculate the corresponding feature vector. Supposing there are M key frames of the query video, and N source videos.

Step2: Using Scalar Quantizer to search the closet video in the source for the query video. Since the unique nearest neighbor search is set, each query frame of the query video has the closest Euclidean distance to the key frame of the corresponding source video. According to the search results, one source video will correspond to M Euclidean distances, namely $X = \{D_1, D_2, \ldots D_M\}$. Now setting a preliminary distance threshold of D_f, then keeping the search distance which is under this distance threshold, and discarding the search distance which is not under this distance threshold, that is:

$$D_j = \left\{D_j \in X \middle| D_j \le D_f\right\}, 1 \le j \le n \tag{9}$$

Step3: Assuming that the number of search distances saved after step2 is n, now setting a ratio threshold of R_f, that is, if it meets:

$$\frac{n}{M} > R_f \tag{10}$$

Then the n searched distances are averaged at the n image frame level, that is:

$$\overline{D} = \frac{D_1 + D_2 + \cdots + D_n}{n} \tag{11}$$

Step4: Since a query video needs to be searched for N cycles in the source, N average Euclidean distances are obtained after distance threshold and ratio threshold processing. And we need to set a matching distance threshold to find the nearest average Euclidean distance.

In fact, the modified algorithm firstly sets a distance threshold on the basis of the original algorithm, selecting the query frame distances which are less than this distance threshold, and recording the number of query frames that meet the conditions. Then we set a ratio threshold to observe whether the ratio of the number of the query frames that meet the conditions to the whole number of query frames of a query video is greater than it, if so, performing the same matching steps as the original algorithm. It should be noted that the distance threshold D_f and the ratio threshold R_f are the most effective ratios among many other ratios we set.

The algorithm 2 diagram is as follows:

	Source A	Source B	Source C	Source D	···	Source N
qF1 🟦	AD1	BD1	CD1	DD1	···	ND1
qF2 🟦	AD2	BD2	CD2	DD2	···	ND2
qF3 🟦	AD3	BD3	CD3	DD3	···	ND3
···	···	···	···	···	···	···
qFn 🟦	ADn	BDn	CDn	DDn	···	NDn
	$\frac{\sum_{i=1}^{n} ADi}{n}$	$\frac{\sum_{i=1}^{n} BDi}{n}$	$\frac{\sum_{i=1}^{n} CDi}{n}$	$\frac{\sum_{i=1}^{n} DDi}{n}$	···	$\frac{\sum_{i=1}^{n} NDi}{n}$

Fig.4 The algorithm 2 diagram

IV. Experiment

This paper proposes a similarity measurement algorithm based on the image frame level to realize video retrieval. We use the deep neural network Resnet18 to extract image features, and then the Scalar Quantizer based on the approximate nearest neighbor search is used for distance search. The similarity of the obtained distance is measured to obtain the best matching video, and finally the recall and the precision are used to measure the match algorithm. It should be noted that the specific algorithm operation of the experiment in this chapter is integrated according to the content of Chapter 3. And the experimental flowchart is shown as follows.

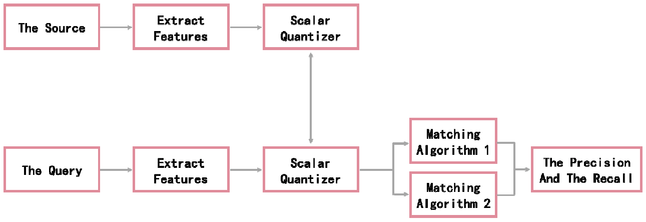


Fig.5 The experimental flowchart

A. Experimental Design

1) Selection of data sets

This paper selects 50 videos with a duration of 1 hour to 2 hours from the Internet as the copyrighted video set, also known as the source. And then performing various video transformations (converting format, changing resolution, adding subtitles and logos, changing video aspect ratio, pirating) on these 50 source videos to form 95 infringements with a duration of 0 to 10 minutes[8]，which called the query A. Then selecting 100 short videos from the Internet which are irrelevant to the source to compare with the query A, and we call them as the query B. For the query A, since the videos in it are all infringing videos corresponding to copyright videos, the ideal result is that each video in the query A can find the corresponding video in the source through the retrieval algorithm. For the query B, since the videos in it are all irrelated to videos in the source, the ideal result is that the retrieval algorithm cannot retrieve the source video corresponding to the videos in the query B.

2) The frame extraction

First of all, this paper uses the deep convolutional network Resnet18 to extract the semantic features of the image frame. The processing of the three data sets must go through the feature extraction, and it includes decoding, frame extraction, and calculation of feature vectors.

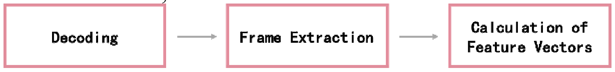


Fig.6 The feature extraction

The frame extraction method adopted in this paper is to extract one frame per second for the videos in the source. For the videos in the query A and the query B, the frames are extracted according to their own time length, that is, if the video is less than 1 minute, one frame is extracted every second, else a total of 60 frames will be extracted in the same time interval. This method is used to maximize the possibility of not discarding any key frames to achieve similarity matching of the video at the image frame level.

B. *Experimental result*

According to the experimental method, setting different matching distance thresholds will result in different precision and recall. To ensure accuracy, we recorded every query video of the query A and its corresponding source video(each query video corresponds to a source video). In the same algorithm, trying to find the source video that can accurately correspond to the query video in the query A and the invalid source video for the query video in the query B under the matching distance threshold, and recording its precision. And the recall is recorded by finding the source video that can accurately correspond to the query video in the query A. We will set different matching distance thresholds at the video level and record the precision and recall of these two algorithms to obtain the retrieval information.

1) *The result of the algorithm 1*

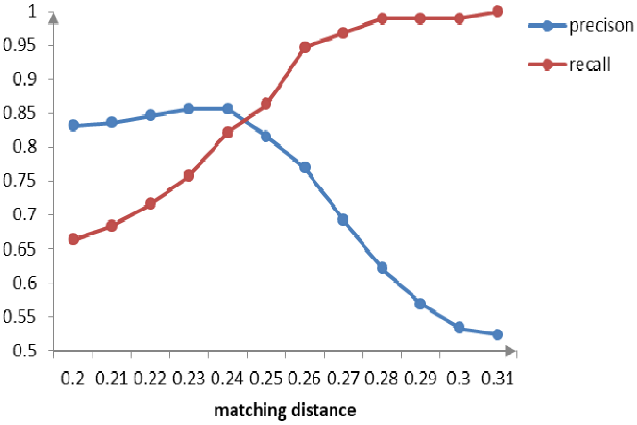


Fig.7 The result of the algorithm 1

2) *The result of the algorithm 2*

It should be noted that the precision and the recall of the algorithm 2 vary with the D_f and the R_f. We found that the number of invalid source videos is increasing with the R_f increasing under the constant D_f and the number of accurate videos is increasing with the D_f decreasing under the constant R_f. Since it has three sets of variables, we set the values of D_f to 0.25, 0.26, 0.27 and 0.28 respectively, and set R_f to 0.1, 0.2, 0.3, 0.4 and 0.5 for each value of D_f through the analysis of the distance results. By observing the matching distance results of these 20 sets of variables, we found that when the value of D_f is 0.26, the most stable matching relationship can be obtained. So we set D_f to 0.26, and observe how the corresponding precision and recall rates change with the matching distance when R_f is 0.1, 0.2, 0.3, 0.4 and 0.5 respectively. Now taking D_f=0.26 and R_f=0.1 as the group A, D_f =0.26 and R_f =0.2 as the group B, D_f=0.26 and R_f=0.3 as the group C, D_f=0.26 and R_f=0.4 as the group D, D_f=0.26 and R_f=0.5 as the group E to observe the results.

a) Setting the D_f=0.26 under the 5 values of the R_f to observe the precision with the change of the matching distance at the video level.

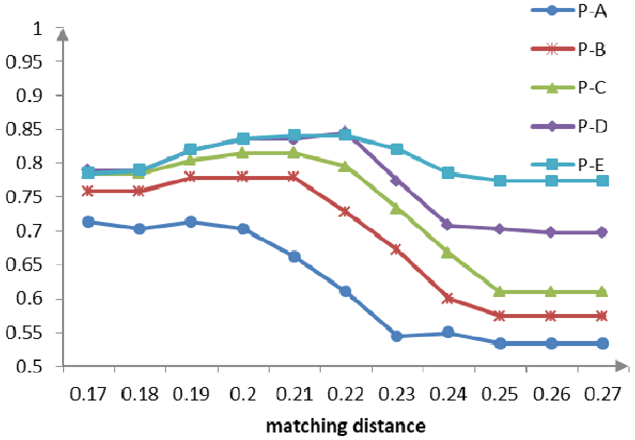


Fig.8 The result of the a)

b) Setting the D_f=0.26 under the 5 values of the R_f to observe the recall with the change of the matching distance at the video level.

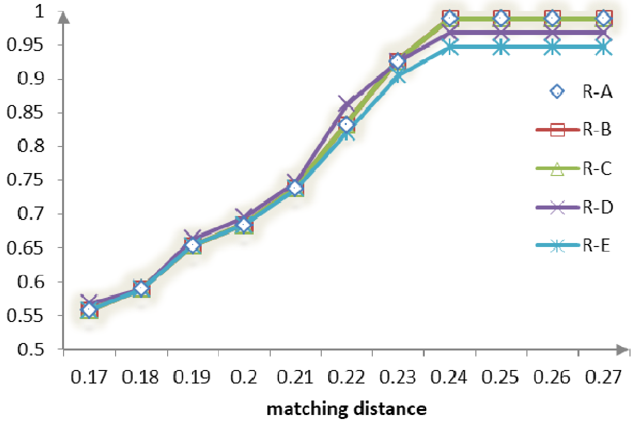


Fig.9 The result of the b)

C. *Experimental conclusions*

It can be seen that with the gradual increasing of R_f, the corresponding precision increases when the matching distance at the video level increases, while the recall remains basically unchanged, which is under the condition of the constant D_f. Taking the correlation between precision rate and recall rate as the measurement standard, we hope that the numerical gap between the two should be small within a certain range, that is, the correlation is good, so that the retrieval performance obtained is suitable for the actual retrieval. Therefore, it can be seen from the result graph that when D_f=0.26 and R_f=0.4, the correlation between precision and recall is best as the matching distance increases. From the actual retrieval effect, when D_f=0.26 and R_f=0.4, and the matching distance is 0.22, the precision is 0.846, and the recall is 0.863, which achieves the best relevance and the best search effect.

It can be seen from the experimental results that the two algorithms have in common:

1) As the matching distance threshold increases, the precision decreases and the recall increases.

2) The precision and the recall are both varied from 0.5 to 1, which are mutually restricted.

3) The lower or higher the matching distance threshold, the corresponding precision and recall rates have no judgment value on the retrieval effect and cannot be used for actual needs.

The differences between the two algorithms are:

1) To reflect the numerical relationship between the precision and the recall betterly, the distribution range of the matching distance threshold of Algorithm 2 at the video level is smaller than that of Algorithm 1.

2) Before the precision and the recall are balanced, the correlation between the two in Algorithm 1 is better; and after the balance, the correlation between the two in Algorithm 2 is better.

Both algorithms have a certain degree of robustness and strong retrieval capabilities. When the matching distance threshold of each algorithm takes the middle range, the comprehensive performance of precision and recall is the best. However, combining the two measurement indicators, in the case of the best correlation, the precision and recall of the two algorithms have not exceeded 0.9, both are 0.8-0.9. The reasons are: 1) The frame extraction cannot be approximated as continuous at the time node. 2) Due to frame extraction and picture distortion, the nearest neighbors may not be the exact same picture[9]. 3) When the query video and the source video have duplicate shots, misalignment and confusion in the matching relationship will be caused.

V. Conclusion and expectation

According to actual needs, in the retrieval of infringing short videos, the idea adopted in this paper is to use deep neural networks for video feature extraction. And then using the Scalar Quantizer to realize the rapid and effective matching among feature vectors.Finally, realizing the nearest neighbor matching of the average Euclidean distance at the image level to find the matched video. This paper proposes two matching algorithms, and evaluates the effect of the algorithm based on the precision and the recall under the same data set. We found that the two algorithms are both robust to the retrieval of infringing videos, but the precision and recall of the two algorithms have not exceeded 0.9 in the case of the best correlation.

In the following scientific research, we will continue to explore video retrieval algorithms to apply to larger video retrieval data sets based on the deficiencies of the existing algorithms and the idea of improving the algorithms.

Acknowledgement

This work was supported by the National Key R&D Program of China (2019YFB1406100) and the Key R&D Program of Shanxi (201903D421007). It was also the research achievement of the Key Laboratory of Digital Rights Services.

References

[1] Y. Jiang , Y. Jiang and J. Wang . “VCDB: a large-scale database for partial copy detection in videos.” European conference on computer vision, 2014, pp.357-371.

[2] X. Zhang, Y. Xie and X. Luan, et al. “Video copy detection based on deep CNN features and graph-based sequence matching.” Wireless Personal Communications. 2018, vol.103, pp.401-416.

[3] Y. Lou, Y. Bai and J. Lin, et al. “Compact deep invariant descriptors for video retrieval.” Data Compression Conference , 2017, pp.420-429.

[4] G. Zhao, M. Zhang and Y. Li, et al. “Pyramid regional graph representation learning for content-based video retrieval.” Information Processing & Management, 2021, vol.58, pp.102488.

[5] A. Babenko and V. Lempitsky, “Aggregating local deep features for image retrieval.” Proceedings of the IEEE international conference on computer vision, 2015, pp.1269-1277.

[6] G. Tolias, R. Sicre and H. Jégou. “Particular object retrieval with integral max-pooling of CNN activations.” International Conference on Learning Representations, 2016, pp.1-12.

[7] Z. Peric, B. Denic and M. Savic, et al. “Design and Analysis of Binary Scalar Quantizer of Laplacian Source with Applications.” Information, 2020, vol.11, pp.501.

[8] Q. Jiang, Y. He and G. Li, et al. “SVD: A large-scale short video dataset for near-duplicate video retrieval.” Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp.5281-5289.

[9] Y. Liu, S. Dhakal and B. Hao, “Multimedia image and video retrieval based on an improved HMM.” Multimedia Systems, 2020,vol.3, pp. 1-11.