Hindawi
Advances in Multimedia
Volume 2020, Article ID 7862894, 8 pages
https://doi.org/10.1155/2020/7862894

Research Article

Large-Scale Video Retrieval via Deep Local Convolutional Features

Chen Zhang[1], **Bin Hu**[2,3], **Yucong Suo**[4], **Zhiqiang Zou**[1,5] and **Yimu Ji**[1]

[1]*College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China*
[2]*College of Geographic Science, Nanjing Normal University, Nanjing, China*
[3]*Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Ministry of Education, Nanjing, China*
[4]*Bell Honor School, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China*
[5]*Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing, Jiangsu, China*

Correspondence should be addressed to Bin Hu; hb_hubin@126.com and Zhiqiang Zou; zouzq@njupt.edu.cn

Received 19 November 2019; Revised 1 February 2020; Accepted 10 February 2020; Published 9 June 2020

Academic Editor: Martin Reisslein

In this paper, we study the challenge of image-to-video retrieval, which uses the query image to search relevant frames from a large collection of videos. A novel framework based on convolutional neural networks (CNNs) is proposed to perform large-scale video retrieval with low storage cost and high search efficiency. Our framework consists of the key-frame extraction algorithm and the feature aggregation strategy. Specifically, the key-frame extraction algorithm takes advantage of the clustering idea so that redundant information is removed in video data and storage cost is greatly reduced. The feature aggregation strategy adopts average pooling to encode deep local convolutional features followed by coarse-to-fine retrieval, which allows rapid retrieval in the large-scale video database. The results from extensive experiments on two publicly available datasets demonstrate that the proposed method achieves superior efficiency as well as accuracy over other state-of-the-art visual search methods.

1. Introduction

Enormous images and videos are generated and uploaded onto the Internet. With a large amount of publicly available data, visual search has become an important frontier topic in the field of information retrieval. There exist several kinds of visual search tasks, including image-to-image (I2I) search [1, 2], video-to-video (V2V) search [3, 4], and image-to-video (I2V) search [5, 6]. Specifically, the well-known I2I visual search can be used for product search, in which relevant images are retrieved by the query image. The V2V search is commonly used for copyright protection, in which video clips are found via a relevant video. The I2V search addresses the problem of retrieving relevant video frames or specific timestamps from a large database via the query image. This technology is relevant for numerous applications, such as brand monitoring, searching film using slides, and searching lecture videos using screenshots.

In this work, we study the specific task of I2V search, which is especially challenging because of the asymmetry between the query image and the video data. Video data can be divided into four hierarchical structures: video, scene, shot, and frame. When considering only the visual content, a video is a sequence of frames displayed at a certain rate (as shown in Figure 1). For example, a video with a frame rate of 30 fps is equivalent to 30 images in one second. The structure of a video means that adjacent frames are highly correlated with each other. To perform large-scale retrieval, we should select representative frames of a video frame sequence to reduce redundant information for further processes. Key-frame extraction, which could represent the salient content and information of the video, is the technique employed to remove redundant or duplicate frames. In this work, we propose a cluster-based key-frame extraction algorithm to summarize the video sequences.

Inspired by the advances in content-based image retrieval (CBIR), we propose to take advantage of the image retrieval techniques to image-to-video search. In CBIR, one of the most challenging issues is the association of pixel-level information with human-perceived semantics. Although some hand-crafted features have been proposed to represent images, the performance of these descriptors is not

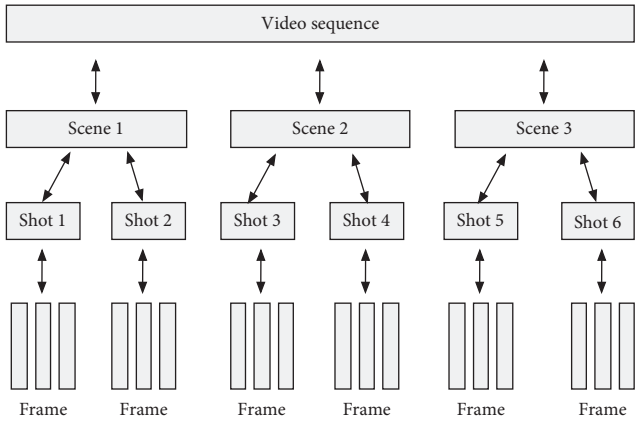


Figure 1: The structure of video data.

satisfactory. Recently, the CNN-based descriptors have shown excellent performance on various computer vision tasks, such as image classification, instant search, and target tracking. Encouraged by the advances in the deep convolutional neural network, our works share similarities with other CNN-based methods extracting features of the frame via pretrained CNNs.

In visual search tasks, search efficiency plays an essential role. Due to the high computational cost, high-dimensional CNN features are not appropriate for large-scale I2V retrieval. To aggregate these high-dimensional features into a lower-dimensional space, we propose a mechanism with two pooling layers for coarse-to-fine search. Specifically, the low-dimensional frame index generated from the second pooling layer is used for the coarse-level search, which could quickly narrow down the matches. And, the high-dimensional frame descriptor generated from the first pooling layer is used for the fine-level search to improve the retrieval accuracy.

This work presents three contributions:

(i) We proposed a cluster-based key-frame extraction algorithm to remove a large amount of redundant information in the video, which could greatly reduce storage cost.

(ii) We took advantage of an aggregation method based on average pooling to encode deep local convolutional features, which allows rapid retrieval in the large-scale video database. To further improve efficiency, we introduced a coarse-to-fine strategy performing the search in two steps.

(iii) An extensive set of experiments on two publicly available datasets demonstrated that the proposed method outperforms several state-of-the-art visual search methods.

2. Related Work

Key-frame extraction is an essential part in video analysis and management, providing a suitable video summarization for video indexing, browsing, and retrieval. The existing key-frame extraction methods are roughly divided into three categories. Early works [7, 8] focused on sampling video sequences uniformly or randomly to obtain key frames, which is easy to implement. However, it ignores the contents of the frames and may result in repeated frames or missing of the important frames. A second generation of works [9, 10] reported significant gains in key-frame extraction based on shot segmentation which selects the key frames from shot fragments. The extracted key frames via this method are representative. However, the neglected correlation between different shots may result in information redundancy. In response to the above problems, cluster-based key-frame extraction [11, 12] has emerged. This method divides the video frame into clusters based on frame contents and then extracts several representative frames from each cluster. The key frames extracted by this method faithfully reflect the original video content. In this paper, we propose a key-frame extraction method based on the k-means clustering algorithm for further processes.

In the image-to-video task, frame representation plays a critical role. In the early 1990s, images were indexed by the hand-crafted features, like color, texture, and spatial. A straightforward strategy for image representation is to extract global descriptors. However, global signatures may fail the invariance expectation to image changes such as illumination, occlusion, and translation. The performance of these visual descriptors was still limited until the breakthrough of local descriptors. In 2003, with the introduction of the Bag-of-Words (BoW) model in the image retrieval community, the majority of the traditional methods were not used any further. For more than a decade, the retrieval community has witnessed the superiority of the BoW model, and many improvements [13, 14] were proposed. In 2012, Krizhevsky et al. [15] proposed AlexNet, which achieved the state-of-the-art recognition accuracy in ILSRVC 2012. Inspired by the advances of deep convolutional neural networks, many works have focused on deep learning-based methods, especially the CNNs. Early works [16, 17] elaborated that features from fully connected layers of a pretrained CNN network perform much better than traditional hand-crafted descriptors. However, several works [18, 19] reported that local features from the last convolutional layer usually yield superior accuracy compared to the global features from the fully connected layer. Our works share similarities with the former methods that extract convolutional features from pretrained CNNs.

However, to perform large-scale retrieval, it is necessary to compress the high-dimensional features to reduce the storage cost and speed up the retrieval. Several works have tried to encode features from CNNs via BoW [20], VLAD [21], and FV [22], which are commonly used to generate hand-crafted descriptors. Although these methods perform well in some visual search tasks, they require a large code book trained offline, which is difficult to achieve in the large-scale database. Additionally, some information will be lost in the feature encoding stage using these methods. Apart from the aggregation strategies mentioned above, average pooling mechanism was able to generate discriminative descriptors. Lin et al. [23] elaborated the reasons why pooling is effective in encoding deep local convolutional features. Firstly, the mean pooling strategy could largely prevent overfitting.

Secondly, it sums up the spatial information, resulting in a more robust spatial transformation of the query image. Inspired by the excellent performance of average pooling, we propose a simple aggregation method to generate compact and discriminative frame representations.

3. Approach

Our method includes three main components: key-frame extraction, frame representation, and coarse-to-fine retrieval, as shown in Figure 2. The first component is a key preprocess to summarize the video data. Subsequently, the feature representation of the key frame is learned by the pretrained deep convolutional neural networks. Ultimately, relevant frames to the query image are retrieved after feature aggregation.

The focus of our work is shown in Figure 2. Figure 2(a) shows the process of indexing and extracting the descriptors for an image, and note that the length of the index is much smaller than that of the descriptor. For large-scale retrieval tasks, it is very important to quickly narrow down the search using the image index. Figure 2(b) shows the process of coarse-to-fine search. In the coarse-level search, the query image's index is compared to the indices of key frames (DB of the index) which are extracted from video frames to generate m candidates. Then, the descriptor of the query image, which contains more information than the index, is compared to the descriptors (DB of the descriptor) of m candidates in the fine-level search using Euclidean distance. The smaller the Euclidean distance is, the higher the level of similarity of the two images is. Each candidate is ranked in an ascending order by similarity; hence, top n ranked frames are selected as the final result.

3.1. Key-Frame Extraction. Key-frame extraction is the basis of video analysis and content-based video retrieval. As mentioned in the previous section, a video is a sequence of frames displayed at a certain rate, and adjacent frames are highly correlated with each other. Key-frame extraction chooses frames to summarize the video while removing redundant information. In this work, we adopt the cluster-based algorithm to extract representative frames.

The main idea of the cluster-based algorithm is to divide the frame sequences into several clusters according to the frame features, and then the frame closest to the cluster center would be selected as a key frame. However, this algorithm requires a prespecified experimental parameter, the number of clusters, which directly affects the result of key-frame extraction. It is very difficult to compute the number of clusters in the case where the video content is uncertain. To address this issue, we propose an improved key-frame extraction algorithm. The specific steps are represented in Algorithm 1, in which steps from (1) to (5) are responsible for computing the number of clusters, while steps from (6) to (9) perform the task of dividing the frame sequences into several clusters and selecting a key-frame sequence.

3.2. Frame Representation. Our approach is similar to former works which extracted convolutional features from pretrained CNNs. However, we discard the softmax and fully connected layers of the original network while keeping convolutional layers to obtain local features. Our work focuses on local features due to the problem that global descriptors may fail the invariance expectation to image changes [24].

In this work, we choose the popular deep neural network named VGG16 to extract frame features, which was trained on the ILSVRC dataset. The network consists of a stacked 3×3 convolutional kernel and max-pooling layers, followed by three fully connected and softmax layers. Table 1 shows the output size of convolutional layers in VGG16. Given a pretrained VGG16 network, an input frame is first rescaled to a predefined image side and then is passed through the network in a forward pass. Finally, we obtain features with size $7 \times 7 \times 512$ from the last max-pooling layer.

3.3. Coarse-to-Fine Retrieval with Aggregated Features. Deep convolutional neural networks have shown their promise as a universal representation for recognition. However, the signatures are high-dimensional vectors that are inefficient in large-scale video retrieval. To facilitate efficient video retrieval, a practical way to reduce the computational cost is to aggregate the CNN features.

Given a frame, we denote the feature map from the last max-pooling layer as f. Assume that f takes the size of $k \times w \times h$, where k denotes the number of channels and w and h are the width and height of each channel. Assume that p represents the output of mean pooling and $s \times t$ ($s \le w, t \le h$) is the pooling window size. Then, we exert mean pooling steps on the local CNN features:

$$p = \frac{1}{s \times t} \sum f(i), \quad i = 1, 2, \ldots, k. \tag{1}$$

Figure 3 depicts the example of encoding the features extracted from the last max-pooling layer before the fully connected layer of the VGG16 network. The given features sized $512 \times 7 \times 7$, and after the first mean pooling process with pooling window sized 7×7, we get the feature descriptor sized $512 \times 1 \times 1$. Then, after the second mean pooling process with pooling window sized 8×1, the feature descriptor is resized to $64 \times 1 \times 1$.

For large-scale retrieval tasks, it is very important to quickly narrow down the search using the feature index. The initial search is computed using the Euclidean distance of the feature index between the query image and the key frames in the database. After that, the top m frames are selected as candidates based on the distance score. Then, to ensure search accuracy, the fine-level search is performed by calculating the distance of the feature descriptor between the query image and the candidates. Finally, top n key frames, a subset of candidates, are picked out.

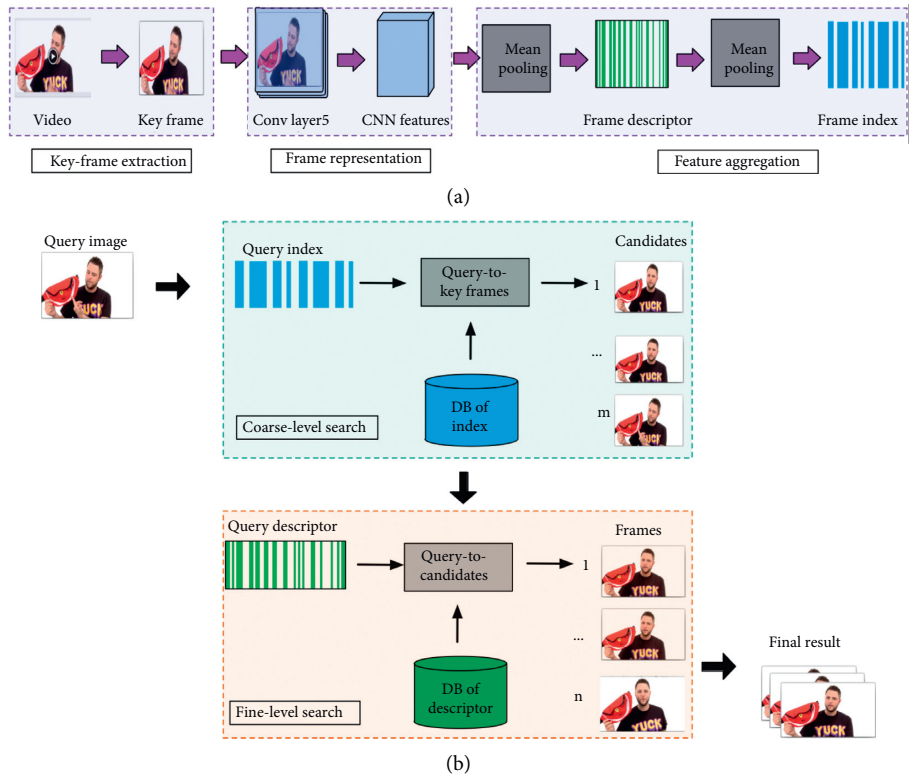


FIGURE 2: Block diagram of our proposed retrieval approach which searches video databases by images. (a) The process of indexing and extracting the descriptors for an image. (b) The process of coarse-to-fine search.

Input: the original video sequence
Output: a key frame sequence $(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_m)$
(1) Split video data into a set of frame sequences $(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_\mathbf{n})$
(2) Calculate the Euclidean distance $(\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_{n-1})$ between adjacent frames according to the color histogram
(3) Calculate the mean distance $\mathbf{D_{ave}} = (\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_{\mathbf{n}-1})/(\mathbf{n}-1)$
(4) Assuming the number of key frames is m, affected by the values of parameters θ and $\mathbf{D_{ave}}$
(5) for $j = 1, \ldots, n-1$ do
  if $(\mathbf{D_j} > \boldsymbol{\theta} \times \mathbf{D_{ave}})$ then
    $m += 1$
  end if
 end for
(6) Select m cluster centers randomly
Repeat
(7) Extract deep convolutional features $(\mathbf{F_1}, \mathbf{F_2}, \ldots, \mathbf{F}_{n-1})$ of video frames via VGG16
(8) Calculate the distance between each frame and the cluster center via the deep convolutional features
(9) Reclassify the corresponding frames according to the minimum distance criterion
(10) Recalculate the cluster center of each class
Until the objects in each cluster no longer change
(11) The cluster center of each class is available, and the frame closest to the cluster center is selected as a key frame

ALGORITHM 1: Key-frame extraction based on the K-means cluster.

4. Experiment

In this section, we demonstrate the benefits of our method. We start with introducing the datasets, evaluation metrics, and parameter setting. Then, we present our experimental results with performance comparison with several existing visual search approaches.

4.1. Experimental Preparation

4.1.1. Datasets. We consider 2 datasets. The NTU video object instance dataset (NTU) [25] and the 2001 TREC video retrieval test collection (2001 TREC) [26]. The NTU consists of 146 video clips from YouTube or mobile cameras. The total size of these clips is 274 MB, and the average duration is 10.54 seconds.

Table 1: Structure of VGG16.

Layer	Output size
Conv3-64	$224 \times 224 \times 64$
Conv3-64	$224 \times 224 \times 64$
Max-pooling	$112 \times 112 \times 64$
Conv3-128	$112 \times 112 \times 128$
Conv3-128	$112 \times 112 \times 128$
Max-pooling	$56 \times 56 \times 128$
Conv3-256	$56 \times 56 \times 256$
Conv3-256	$56 \times 56 \times 256$
Conv3-256	$56 \times 56 \times 256$
Max-pooling	$28 \times 28 \times 256$
Conv3-512	$28 \times 28 \times 512$
Conv3-512	$28 \times 28 \times 512$
Conv3-512	$28 \times 28 \times 512$
Max-pooling	$14 \times 14 \times 512$
Conv3-512	$14 \times 14 \times 512$
Conv3-512	$14 \times 14 \times 512$
Conv3-512	$14 \times 14 \times 512$
Max-pooling	$7 \times 7 \times 512$

The second dataset consists of 11 hours of the publicly available MPEG-1 video provided by the TREC conference series. We experiment with 2G video clips, a subset of 2001 TREC, to evaluate the performance of our approach.

4.1.2. Evaluation Metric. Query images for retrieval are captured by OpenCV, an open source library for computer vision. For evaluation, it is considered a visual match on condition that the query image and the retrieved frame are from the same video clip. Performance is measured in terms of accuracy:

$$\text{Acc} = \frac{\text{no. (visual matches)}}{\text{no. (retrieved frames)}}. \quad (2)$$

In order to show the performance variation, we test different parameter settings for our key-frame extraction algorithm. There is one parameter to be tuned in our proposed model: θ. The compression ratio is used to measure the compactness of the extracted key-frame sequence, which is defined as

$$\text{compression ratio} = 1 - \frac{\text{no. (key frames)}}{\text{no. (frames)}}. \quad (3)$$

4.1.3. Parameter Setting. Figure 4 shows the compression ratio and retrieval accuracy variations with varying θ. When the value of θ is less than 2, the compression ratio improves dramatically with the increase in θ. After that, the compression ratio keeps steady and infinitely close but no more than 1. The higher the compression ratio, the more the redundant frames are lost and the more the storage space is saved. The accuracy keeps steady when the value of θ is less than 1.4. After that, the accuracy drops dramatically with the increase in θ.

The accuracy is based on smaller θ. However, it also leads to a lower compression ratio, which will decrease the memory efficiency. We set the final value of θ to 1.4 by making a tradeoff between accuracy and efficiency. The summary of the information for the two datasets is shown in Table 2.

4.2. Experimental Results. To evaluate the performance of our proposed coarse-to-fine search method, we compare with several existing visual search approaches, which are briefly described as follows:

(i) *Deep Feature-Based Method (DF).* Babenko et al. [16] introduced features of pretrained CNN for image classification to replace traditional hand-crafted descriptors. We use the deep convolutional features from the last convolutional layer of VGG16 as a baseline method.

(ii) *Deep Feature Spatial Encoding (DFSE).* Perronnin et al. [27] focused on encoding the deep convolutional features of CNN using the FV to generate frame descriptors.

(iii) *Deep Feature Temporal Aggregation (DFTA).* Noa et al. [28] proposed to aggregate the deep convolutional features of all frames within one shot via max-pooling. In DFTA, features in the same shot are aggregated into a single feature to reduce redundant information between adjacent frames.

(iv) *Local Binary Temporal Tracking (LBTT).* LBTT [28] is based on the summarization of hand-crafted local binary features, which encode the pixel intensity value of frames into 256-dimensional binary vectors.

(v) *Deep Feature Spatial Pooling (DFSP).* To evaluate the performance of the pooling strategy, we used the 64-dimensional index of the frame for retrieval, which is generated after two pooling layers.

All the experiments are implemented on a computer which has Inter Core i5 2.3 GHz 2 processors, 8 GB RAM, and macOS 10. Tables 3 and 4 show the examples of our retrieval results on the two datasets.

4.2.1. Results on the NTU Dataset. We first test different methods on the NTU. The accuracy, search time, and frame descriptors' dimension of different methods are presented in Table 5. Our method involves a coarse-to-fine retrieval process. In the coarse search, the dimension is 64, and in the fine search, the dimension is 512, which are described in the first line in Table 5. The proposed method achieves the best results in terms of accuracy, improving the performance by 0.05 compared to DF. DFSP and DFSE consume the shortest time without taking into account the time spent in offline training. This is probably because these frame descriptors are 64-dimensional, lower than that of the other methods. To further test the impact of the frame descriptors' dimension on the retrieval speed, we experiment on the large-scale dataset, 2001 TREC. The results of different methods are shown in Table 6.

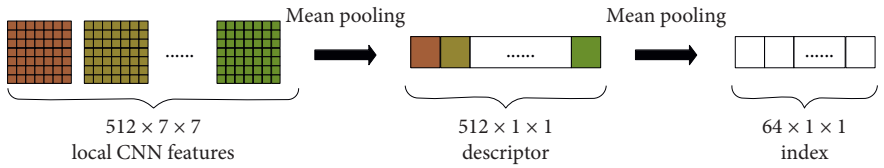


Figure 3: The process of feature encoding.

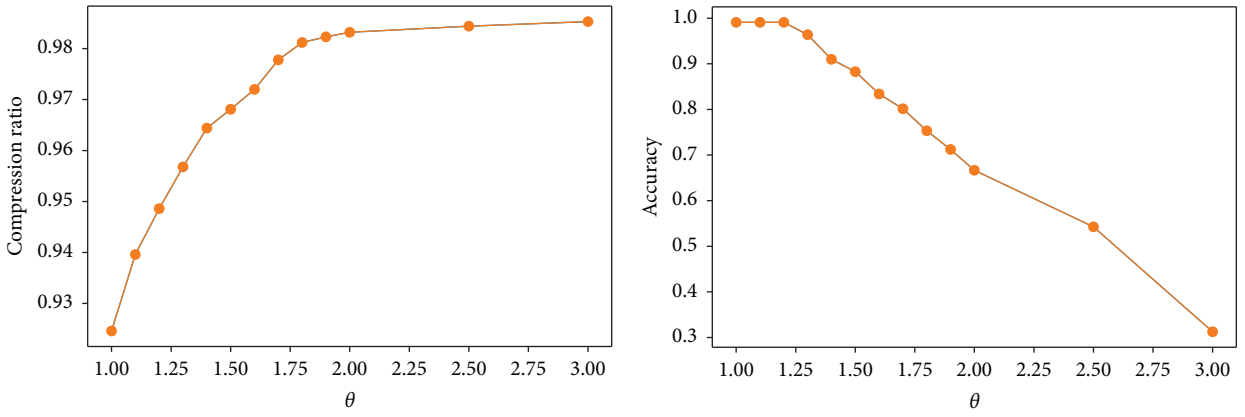


Figure 4: The compression ratio variations of different θ.

Table 2: The information of two datasets used for experiments.

Dataset	The NTU	The 2001 TREC
Size	274 MB	2G
Average duration	10.54 seconds	2.86 minutes
Number of frames	12359	544275
Number of key frames	440	19275
Compression ratio	96.44%	96.45%

Table 3: Example of the top 12 similar frames for the query image on the NTU.

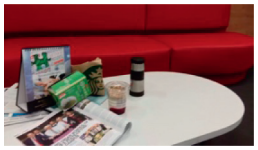
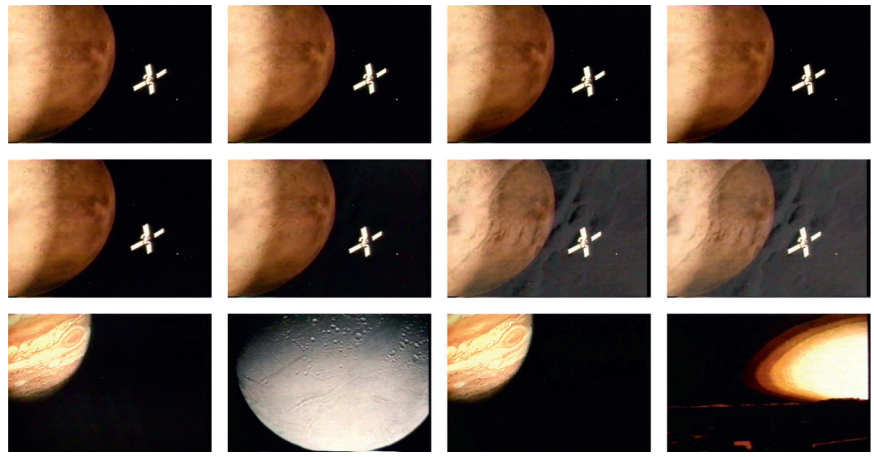
Query image	Top 12 similar frames
 The NTU	

Table 4: Example of the top 12 similar frames for the query image on the 2001 TREC.

Query image	Top 12 similar frames
 The 2001 TREC	

4.2.2. Results on the 2001 TREC Dataset. From Table 6, we can see that the accuracy of all methods is slightly reduced and search time is much longer compared to Table 5. The meaning of the dimension in Table 6 is similar to that in Table 5. For example, the dimensions in our method are 64 and 512, respectively. Our proposed method achieves the best results in terms of accuracy and outperforms other methods by large margins. Note that the accuracy for the proposed method is 0.9153 while that for DFTA is 0.7856. The search time of our method is slightly longer than that of DFSP because it takes time for fine-level search. Although the retrieval speed is slightly reduced, the retrieval accuracy is greatly improved. Therefore, we believe that our proposed coarse-to-fine search is effective. The accuracy of DF is worse than that of DF and DFSP. Furthermore, its search time is about 2-3 times longer than DFSP. This shows that the pooling strategy is effective in encoding deep local convolutional features. However, the accuracy of DFSE and DFTA is worse than DF although the search time is shorter. This indicates that although high-dimensional descriptors could be encoded into a lower-dimensional space via these two methods, they could lose a lot of feature information during the encoding process.

Table 5: Comparison with existing visual search approaches on the NTU.

Method	Acc	Time (s)	Dimension
Ours	**0.9691**	**1.67**	**64, 512**
DFSP	0.9516	1.613	64
DF [16]	0.9198	1.892	25088
DFSE [27]	0.8441	1.606	64
DFTA [28]	0.8254	1.739	512
LBTT [28]	0.9096	1.693	256

Table 6: Comparison with existing visual search approaches on the 2001 TREC dataset.

Method	Acc	Time (s)	Dimension
Ours	**0.9213**	**5.302**	**64, 512**
DFSP	0.8841	5.164	64
DF [16]	0.8313	14.305	25088
DFSE [27]	0.8106	5.201	64
DFTA [28]	0.8027	8.132	512
LBTT [28]	0.8174	6.764	256

5. Conclusion and Future Work

In this paper, we proposed a method based on deep local features to solve the problem of image-to-video retrieval. The models presented in this work are based on key-frame extraction and feature representation. The experimental results demonstrated that our method achieved competitive performance with respect to other CNN-based representations, as well as performed excellent in the cost of indexing and search time.

However, the proposed method appears to be more appropriate for tasks in which query images are from the original video frames. The quality problem of the query image caused by geometric transformations and occlusion might affect the search accuracy. In future work, we aim to explore an effective method to reduce the impact of image quality issues.

Data Availability

All data generated or analyzed during this study are included in this paper. The datasets used in this paper, the NTU video object instance dataset, and the 2001 TREC video retrieval test collection can be downloaded from https://sites.google.com/site/jingjingmengsite/research/ntu-voi/data and https://open-video.org//.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of PR China (no. 41571389), the National Key R&D Program of China (2017YFB1401302 and 2017YFB0202200), the Key Laboratory of Spatial Data Mining & Information Sharing of the Ministry of Education, Fuzhou University (no. 2016LSDMIS07), and the Technology Innovation and Application Demonstration Major Program of Chongqing (cstc2018jszx-cyztzxX0015).

References

[1] K. Lin, Y. Huei-Fang, H. Jen-Hao, and C. Chu-Song, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Boston, MA, USA, June 2015.

[2] M. A. Marzouk, "Improving web based image retrieval with fuzzy descriptors relevance feedback technique," *Journal of Computers*, vol. 28, no. 3, pp. 11–26, 2017.

[3] S. Poullot, T. Shunsuke, N. Anh Phuong, and J. Hervé, "Temporal matching kernel with explicit feature maps," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ACM, Brisbane, Australia, October 2015.

[4] S. R. Shinde and G. G. Chiddarwar, "Recent advances in content based video copy detection," in *Proceedings of the International Conference on Pervasive Computing (ICPC)*, IEEE, Pune, India, January 2015.

[5] A. Araujo, "Large-scale query-by-image video retrieval using bloom filters," 2016, http://arxiv.org/abs/1604.07939.

[6] A. Araujo, "Temporal aggregation for large-scale query-by-image video retrieval," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, IEEE, Quebec City, QC, Canada, September 2015.

[7] H. J. Zhang, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition*, vol. 30, no. 4, pp. 643–658, 1997.

[8] Xu-D. Wu, L. Tie-Yan, L. Kwok-Tung, and F. Jian, "Dynamic selection and effective compression of key frames for video abstraction," *Pattern Recognition Letters*, vol. 24, pp. 9-10, 2003.

[9] Li-J. Qin, Z. Yue-Ting, W. Fei, and P. Yun-He, "An integrated framework for shot boundary detection with multi-level features similarity," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, IEEE, Shanghai, China, China, August 2004.

[10] H. Aoki, S. Shimotsuji, and O. Hori, "A shot classification method of selecting effective key-frames for video browsing," in *Proceedings of the Fourth ACM International Conference on Multimedia*, ACM, Boston, MA, USA, Febuary 1997.

[11] S. E. F. De Avila, A. P. B. Lopes, and A. de Albuquerque Araújo, "VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[12] R. da Luz, B. Qin, and T. Liu, "A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2375–2384, 2012.

[13] E. Mohedano, S. Amaia, M. Kevin, M. Ferran, N. E. O'Connor, and N. Xavier Giro-i, "Bags of local convolutional features for scalable instance search," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ACM, New York City, NY, USA, April 2016.

[14] G. Csurka, R. D. Christopher, F. Lixin, W. Jutta, and B. Cédric, "Visual categorization with bags of keypoints," in *Proceedings of the Workshop on Statistical Learning in Computer Vision*, Meylan, France, 2004.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 1, pp. 1097–1105, 2012.

[16] A. Babenko, "Neural codes for image retrieval," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, September 2014.

[17] A. Sharif Razavian, A. Hossein, S. Josephine, and C. Stefan, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, June 2014.

[18] H. Noh, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 2017.

[19] Y. Kalantidis, C. Mellina, and O. Simon, "Cross-dimensional weighting for aggregated deep convolutional features," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, September 2016.

[20] P. Kulkarni, Z. Joaquin, J. Frederic, P. Patrick, and C. Louis, "Hybrid multi-layer deep CNN/aggregator feature for image classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Brisbane, QLD, Australia, April 2015.

[21] Y. Gong, W. Liwei, G. Ruiqi, and L. Svetlana, "Multi-scale orderless pooling of deep convolutional activation features," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, September 2014.

[22] H. Jégou, D. Matthijs, S. Cordelia, and P. Patrick, "Aggregating local descriptors into a compact image representation," in *Proceedings of the CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, San Francisco, CA, USA, June 2010.

[23] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, http://arxiv.org/abs/1312.4400.

[24] J. Richiardi, H. Ketabdar, and A. Drygajlo, "Local and global feature selection for on-line signature verification," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, IEEE, Seoul, Korea, September 2005.

[25] J. Meng, "Object instance search in videos via spatio-temporal trajectory discovery," *IEEE Transactions on Multimedia*, vol. 18, no. 1, pp. 116–127, 2015.

[26] E. M. Voorhees, "Overview of the TREC 2001 question answering track," TREC, 2001.

[27] F. Perronnin, L. Yan, S. Jorge, and P. Hervé, "Large-scale image retrieval with compressed Fisher vectors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, CA, USA, June 2010.

[28] N. Garcia, "Temporal aggregation of visual features for large-scale image-to-video retrieval," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ACM, Yokohama, Japan, June 2018.