Compact CNN Based Video Representation for Efficient Video Copy Detection

Ling Wang, Yu Bao, Haojie Li(✉), Xin Fan, and Zhongxuan Luo

Dalian University of Technology, Dalian, China
hjli@dlut.edu.cn

Abstract. Many content-based video copy detection (CCD) systems have been proposed to identify the copies of a copyrighted video. Due to storage cost and retrieval response requirements, most CCD systems represent video contents using sparsely sampled features, which tends to lose information to some extend and thus results in unsatisfactory performance. In this paper, we propose a compact video representation based on convolutional neural network (CNN) and sparse coding (SC) for video copy detection. We first extract CNN features from the densely sampled video frames and then encode them into a fixed length vector via the SC method. The proposed representation presents two advantages. First, it is compact while is regardless of the sampling frame rate. Second, it is discriminative for video copy detection by encoding the densely sampled frames' CNN features. We evaluate the performance of proposed representation on video copy detection over a real complex video dataset and marginal performance improvement has been achieved as compared to state-of-the-art CCD systems.

Keywords: Video copy detection · Convolutional neural network · Sparse coding · Video level representation · Dense sampling

1 Introduction

A copy is a duplicate segment of video derived from another video, by means of various transformations. The task of content-based video copy detection is to determine if a given video (query) has its copy in a set of testing videos. Analyzing and comparing the features between the querying and testing videos are the usual methods to cover this task. Copy detection has a wide range of potential applications such as copyright control, business intelligence, etc., thus has attracted lots of research efforts over the last decade [1,20].

The duplicate segments may be as long as the origin video or even shorter than 1 s with some distortions. These distortions include simulated camcording, picture in picture, insertions of pattern, compression, etc. [16]. Variety distortions bring great challenges to the copy detection problem. To address these challenges, TREC Video Retrieval Evaluation (TRECVID) released a content-based copy detection benchmark with a large collection of synthetic queries. It launched

L. Amsaleg et al. (Eds.): MMM 2017, Part I, LNCS 10132, pp. 576–587, 2017.
DOI: 10.1007/978-3-319-51811-4_47

the CCD competition task from 2008 to 2011 [16]. Many solutions have been proposed to tackle this problem. Among them, the most popular way is to use local features like SIFT [11] to match and find similar frame pairs, followed by temporal alignment [3]. In these methods, inverted files and Bag-of-words (BoW) representation are widely adopted for fast frame matching. Actually some of them have achieved near-perfect performance [16].

As a simulated dataset, the TRECVID can not accurately reflect real copy videos. There are more complicated visual transformations and more complex temporal structures in real copy videos. The performance of current state-of-the-art approaches is far away from satisfactory for real video copy detection, which remains CCD still an open issue to the multimedia research community as Jiang et al. evaluated [8].

In order to avoid unaffordable computational and storage burdens, most of the current copy detection works extract video features on sparsely sampled frames, e.g. sampling one or two frames per second [3,8]. We argue that sparse sampling could miss much useful information as most frames are dropped. Thus, if we can encode more information of a video segment, the detection performance will be improved. Meanwhile, deep learning approaches, especially convolutional neural network (CNN) has recently shown powerful ability in extracting distinctive image or object features. It achieved great success in general image classification, object detection tasks and semantic analysis [10,12,24]. Jiang's initial attempts [9] of using CNN features for copy detection also demonstrated its promising advantages over existing traditional methods.

Motivated by the above observations, we propose a novel video level representation which encodes the CNN features of densely sampled frames of a short time video into a compact descriptor, for real video copy detection. We first extract the CNN full-connect (fc) layer features for the sampled frames of a short video and then reduce the features dimension by using PCA. After that, sparse coding method is adopted to sparsely assign each frame feature into a set of M codes. So we can get an M-dimensional sparse vector for each frame. After these steps, a max-pooling operation is performed on each component of these vectors. A compact video level representation is finally derived.

The contributions of this paper are two-fold. (1) The proposed novel video level representation is compact which needs less storage and enables fast retrieval of similar video clips. Its dimension is independent to the sampling frame rate. (2) By encoding the densely sampled frames CNN features, the proposed copy detection method significantly outperforms state-of-the-art traditional approaches; it also achieves competitive precision to lately CNN based method [9] but improve the recall rate about 10%, which is more practical to some applications such as web video monitoring and tracking.

The reminder of this paper is structured as follows. Section 2 briefly reviews some well-known video copy detection works and approaches. In Sect. 3, we introduce our method for video feature representation and copy detection. Experimental results are presented and discussed in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related Work

Video copy detection has attracted a lot of research interests in recent years. The approaches to this task mainly include two categories: local feature based methods and global feature based methods. Local feature based methods are widely leveraged by most works. For instance, LBP is used as visual feature in [15], LBP-CS [5] and SIFT are used by Douze et al. [3]. In [3], the local features are clustered into visual words and the image is represented with a BoW model. Then inverted file is adopted to index images for fast retrieval. In [6], hamming Embedding (HE) is employed to further divide the clusters into sub-spaces to improve the accuracy of local feature matching. Meanwhile, weak geometry consistency (WGC) [6] is used to help eliminating the wrong matching between features. Zhao et al. [25] adopt BoW, HE and WGC to effectively annotate web videos via near-duplicate video detection. All these above works extract local features on sparsely sampled frames for the purpose of low memory usage and efficiently retrieval. However, since one single video frame can produce about one thousand local features, it is still a heavy burden to storage and retrieve the large number of frames' features of a very large video dataset.

As a solution, employing global features could significantly decrease the number of features. Wu et al. [22] extract color histogram as frame features to detect similar videos. In [4], a Fisher Vector (FV) [13] alike representation is proposed to aggregate local features to a global feature. Meanwhile, CNN has shown their absolute advantages in image representation and high speed processing. In [9], Jiang and Wang sample frames at fixed time interval and extract CNN based features for each sampled frame. There are two ways to implement CNN features in his work. One of them is the standard CNN which uses Caffe [7] toolkit with AlexNet [10]. A 4,096-dimensional fc feature is extracted to present a frame. The other way extracts local image patch features using a supervised CNN structure called Siamese convolutional neural network (SCNN). Then image patches are described by features which are ranged from 64-dimensional to 512-dimensional. All these features are organized into a fast retrieval structure to do the match processing. Finally the matching results are aligned to the original videos by temporal network [17] according to their temporal consistency. Comparing to the traditional Hough Voting Alignment, Jiang [8] proves that temporal network is more suitable for temporal alignment.

As can be summarized from the above works, most approaches extract features on sparsely sampled frames rather than densely sampled frames. Much useful information is given up to make a heavy concession for considering the memory usage and efficiency retrieval. We will show in the experiments that this will cause the degrade of performance to some extend due to information loss. To overcome this disadvantage, our target is to find a representation which could aggregate more information and keep the final representation compact and discriminative.

3 Compact Video Representation

We will present the proposed compact video representation for short time video clips in detail. The outline of our video copy detection system is shown in Fig. 1, which is divided to three main steps: frame feature extraction, video feature encoding and video segment matching. Video feature encoding includes compression and aggregation. In this paper we pay more attention to the first two steps which are marked by blue arrows in Fig. 1. The matching step (i.e., fast retrieval and temporal alignment) will be introduced briefly.

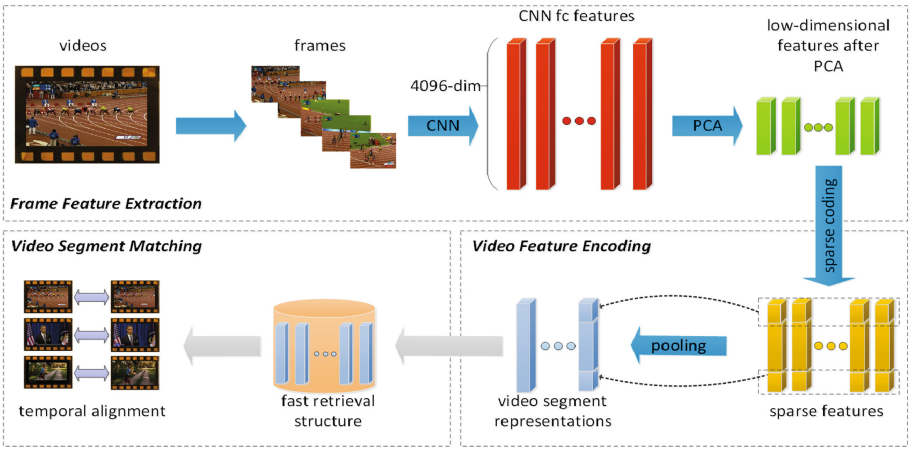

Fig. 1. The outline of our video copy detection system which includes three parts: frame feature extraction, video feature encoding and matching.

3.1 Frame Feature Extraction

Traditional works sparsely sample frames from videos for keeping balance between accuracy and efficiency. On the contrary, our starting point is gathering more information into final representations. So the first step of our method is to sample frames from query and database videos densely.

The next step is to describe sampled frames via frame-based features. Different from local feature extraction processing, CNN has shown its fast processing speed by utilizing parallel GPUs. Although the number of frames becomes more than ten times than the sampling methods in [3,8], the processing time may be equal to or less than the previous methods due to the performance of GPUs. We utilize the Caffe toolkit [7] to implement deep learning algorithm on the densely sampled frames. According to [23], we use the deeper network architecture, i.e. VGG-16layers, which is the winner of VGG ILSVRC 2014 classification task [14]. This network contains 16 weight layers: 13 convolutional layers and 3 fully-connected layers. And other five max-pooling layers are inserted after some convolutional layers.

Finally, we conduct PCA-whitening on the CNN fc features since the dimension of 4,096-D fc features are too high. Up to now, a frame is represented by a low-dimensional feature vector x with length k. We will explore the appropriate value of k in the experiments then.

3.2 Video Feature Compression and Aggregation

We aim to use compact representation to describe the densely sampled features. Directly calculating the distance between frame features is very sensitive to the noise in visual feature and the result is easily influenced by even one dimension of noise feature [18]. Since sparse coding, which models data vectors by the sparse linear combinations of the basis dictionary, could reserve the main components of vectors and make it possible to compactly represent the vectors. There are several works use SC and gains great performance [19]. We compress the frame features using sparse coding in our method. Sparse coding [2] can be regarded as

$$\begin{aligned} &\min_{D,s(i)} \textstyle\sum_i \left\| Ds^{(i)} - x^{(i)} \right\|_2^2 + \lambda \left\| s^{(i)} \right\|_1 \\ &\text{s.t. } \left\| D^{(m)} \right\|_2^2 = 1, \forall m = 1, 2, \ldots, M \end{aligned} \tag{1}$$

where D indicates the overcomplete dictionary, i.e. $M > k$, k is the dimension of feature x. In this work, we set M be four times of k unless otherwise noted. s is the target sparse representation.

We investigated some solutions of sparse decomposition problem and found that the Orthogonal Matching Pursuit (OMP) [21] is the most suitable method for our processing:

$$\begin{aligned} &\min_{D,s(i)} \textstyle\sum_i \left\| Ds^{(i)} - x^{(i)} \right\|_2^2 + \lambda \left\| s^{(i)} \right\|_1 \\ &\text{s.t. } \left\| D^{(m)} \right\|_2^2 = 1, \forall m = 1, 2, \ldots, M \\ &\text{and } \left\| s^{(i)} \right\|_0 \leq T, \forall i \end{aligned} \tag{2}$$

In this case, s could have at most T non-zeros items. We employ K-SVD to train the dictionary. According to the greediness of OMP, the number of non-zeros in s could be close or equal to T. This property is helpful to us while other sparse decomposition methods may make our final features become excessively sparse, which is insufficient to distinguish different video segments.

At this stage, each frame is essentially a sparse feature. The extracted sparse features are then pooled and aggregated into a compact representation for the specified length of video segment. Different from event detection and video classification, video copy detection task needs fine-grained time interval representation for accurately aligning the time line between copies and original video. As shown in Fig. 2, we take 1 s interval in our method.

Video pooling could be divided to three categories: max-pooling, mean-pooling and sum-pooling. Since sparse coding describe a feature by linear combination of its principal basis, it is better to conserve the maximal item among a short time video. Thus, keeping the component-wise maximum by max-pooling

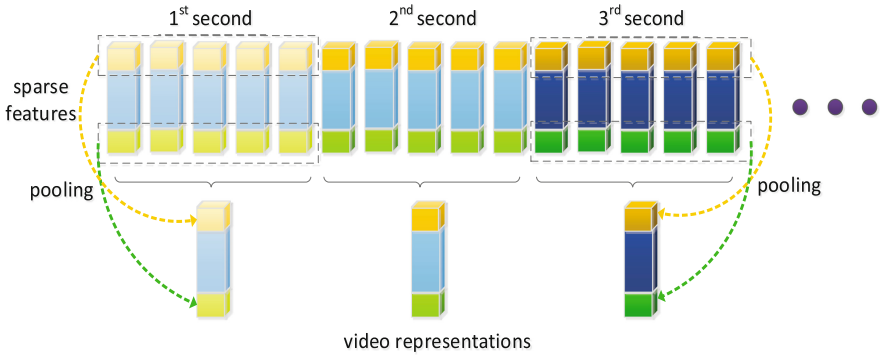

Fig. 2. Illustration of video pooling

is our choice. Because there are some negative items in the sparse feature by the OMP algorithm, we make a small adjustment for max-pooling:

$$\begin{aligned} & v^m = s_i^m, \forall m \in M \\ & \text{s.t.} \ \max_i abs\left(s_i^m\right), i \in 1, 2, \ldots, n \end{aligned} \tag{3}$$

where v is the video representation with the dimension M. s_i is the sparse vector for the i-th sampled frame of the one-swcond interval clip. n is the total number of sampled frames among the one-second interval clip. By this way, we can reserve the most important part of each component whether or not it is positive.

After pooling and aggregating, we finally get a series of video level representations, each of which represents a one-second interval time video segment.

3.3 Video Segment Matching

The final step of video copy detection is to compare the query video with database videos and identify the most similar segment pairs between them. The matching method contains fast retrieval and temporal alignment.

We leverage the normal KD-tree to store our features and do fast retrieval. Although we don't specially investigate its efficiency, there is huge potential of our sparse video features. First, the sparse feature needs less storage and could make it possible to reside all features in memory. Second, a large number of floating point arithmetic is no more needed due to the large amount of matching between zero and zero.

It's necessary to link our video level representations to a longer video segment because each feature only describes a short-time interval clips. The longer video segment is then associated with a starting timestamp and a ending timestamp of a testing video. Following [8], we employ the temporal network to align our matched video segments and adopt the following formula to measure the similarity score between two features:

$$score = e^{-dis^2} \tag{4}$$

where dis is the Euclidean distance between these two features.

4 Experiments

4.1 VCDB Dataset

In our experiments, we utilize the latest released copy detection dataset, namely VCDB dataset [8] to evaluate the proposed method. The VCDB includes core-dataset and distraction-dataset which are all collected from two video-sharing websites: YouTube and MetaCafe. The core-dataset contains over than 500 videos with 9,236 partial copies and the distraction-dataset includes 100,000 videos. We mainly evaluate our video level representations in the core-dataset. Same to the baseline method in [8], all the segments of the 9,236 pairs are considered as a query. If both the two segments among a detected video pairs had intersection time with a ground-truth pair, they would be considered as a correct pairs in spite of the length of overlapped time window. Because a video segment with one single copy frame can be fully demonstrated as a copy pair. We use the precision and recall to measure our features' performance:

$$precision = \frac{|\text{correctly retrieved segments}|}{|\text{all retrieved segments}|} \tag{5}$$

$$recall = \frac{|\text{correctly retrieved segments}|}{|\text{ground-truth copy segments}|} \tag{6}$$

4.2 Experimental Results and Comparisons

We show the results of the proposed method and also compare it with some state-of-the-art systems from several aspects. Our method achieves the best results with the following settings: (1) all frames are used to generate our final representation without sampling; (2) the dimension of features at the frame feature extraction step is set to be 512-D, which results in a 2,048-D video segment representation; (3) the number of non-zero components in the sparse representation is controlled to be at most 32. These settings are both utilized on the fc6 and fc7 layers of VGG-16layer network. We adjust the threshold of the segment pairs' matching score to draw the precision-recall curves.

The comparing of our method, the baseline system [8], standard CNN and SCNN [9] are shown in Fig. 3. As we can see, our methods achieve remarkable performance both on fc6 and fc7, while features extracted based on fc6 works better than on fc7. This may suggest that fc6 is more suitable than fc7 for video copy detection task.

The green curve represents the baseline method which is proposed in [8] and utilizes local features, i.e., SIFT and temporal network to detect copy pairs. This approach is widely used by previous work and has shown near-perfect performance in TRECVID benchmark, however, it is far away from satisfactory in real complexity copy detection.

The red curve shows the result of standard CNN method [9] that extracts features using CNN with the AlexNet and directly uses the 4,096-D features on fc6 for retrieving. It also proposed a fusion method by combining SCNN and

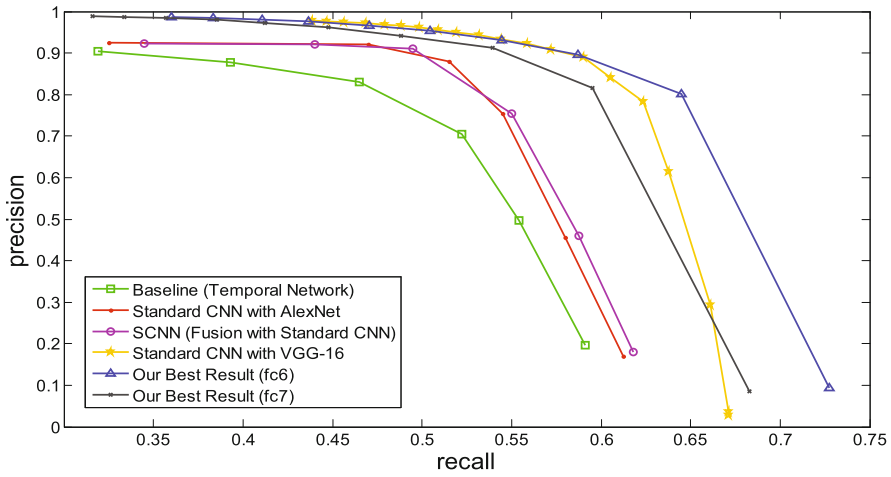

Fig. 3. Precision-recall curves for different methods on the core-dataset of VCDB. (Color figure online)

standard CNN to extract frame features in [9]. Our method performs better than these two CNN based methods, owing to the densely sampled frames feature with more information and the deeper network with better descriptive power.

To demonstrate the effectiveness of dense sampling strategy and compact video feature representation, we also implemented standard CNN method on fc6 using the VGG-16layers network and show its results with the orange curve in Fig. 3. Comparing to it, our method (blue curve) significantly increases the recall rate while maintains a good precision rate, which is more practical to some applications such as web video monitoring and tracking where low miss is more important.

4.3 Impacts of Parameters

Sparsity. We investigated several sparse decomposition algorithms and found that the sparsity of our features obviously affects the performance. We employ the OMP algorithm in our experiments due to the controllable sparsity, and adjust the number of non-zero components in the sparse frame features to produce different sparsity of final video representations. In Fig. 4, the parameter T indicates the maximum number of non-zero components in a sparse frame feature. We conduct experiments on different T while fix the rest parameters to see the performance changes with the sparsity of our representation. As can be seen from Fig. 4, fc6 performs better than fc7 and T exhibits little influence on fc7. For fc6, the larger value of T results in higher performance. However, the performance improves slowly when the value of T increases 32 from 16. Therefore, we set T to 32 in all our experiments.

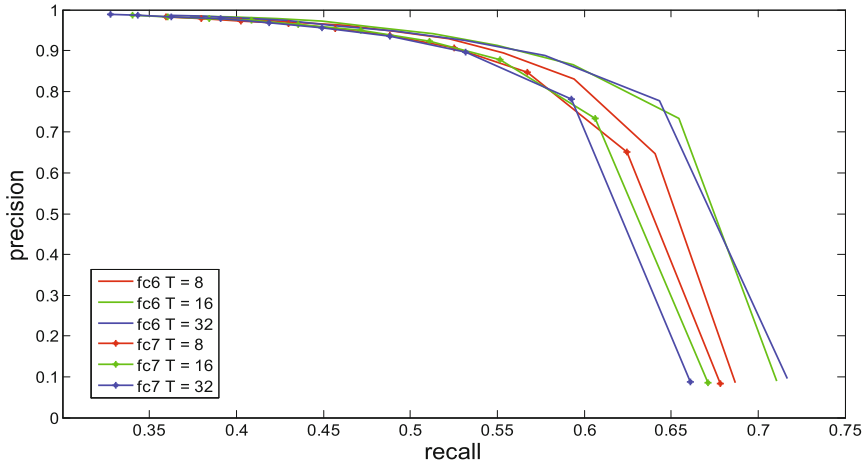

Fig. 4. Results comparison for different sparsity of features

Dimension. The dimension of features after PCA-whitening could influence the dimension of our final representations. It's essential to investigate the impact of the dimensions on the final detection results. In Fig. 5, only the dimension of PCA is changed. From the figure, we can see that the performance is increasing with the rise of dimension. However, the storage cost and retrieval time will both growing exponentially. In our method, we adopt 512-D as a trade-off of performance and resource consumption, which leads to a 2,048-D final video representation.

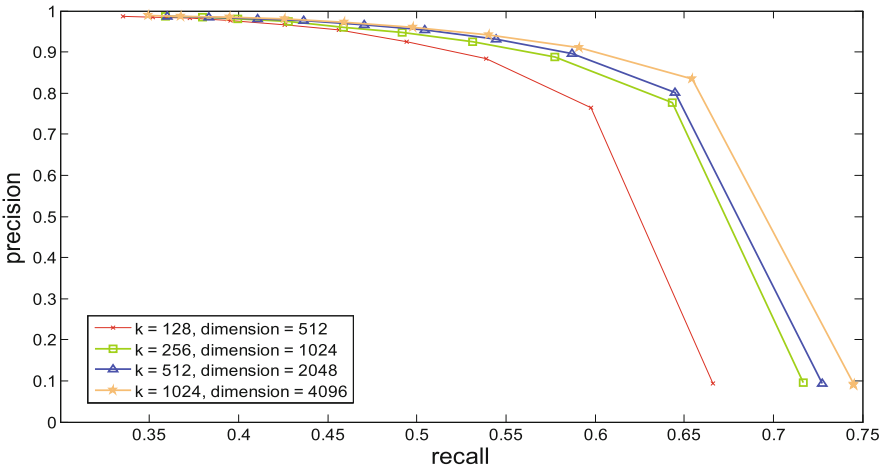

Fig. 5. Results comparison for different dimension of features

4.4 Analysis of the Impact of Sampling Rate

We are the first to utilize densely sampled frames to generate video representations for video copy detection, it is necessary to investigate whether more frames bring better discrimination. To do so, we test the proposed method on fc6 with different sampling rate in the pooling step. The best F1-measure (i.e., the harmonic mean of precision and recall) is used to evaluate the performance.

Table 1. Detection results with different sampling rate of frames

1-frame per sec	2-frames per sec	1/5 frames	1/3 frames	1/2 frames	All frames
0.6695	0.6879	0.6994	0.6995	0.7026	**0.7038**

Table 1 shows that the detection performance increases consistently with the sampling rate, and when the sampling rate increases to 1/5, the performance becomes relatively steady. This is reasonable because the adjacent frames are extremely similar and contain much redundant information. However, 1/5 of 1 s video means 5 to 6 frames, which is denser than any existing methods and could bring much more storage cost and much longer retrieval time to these methods. By fusing the sparse coding and max-pooling strategies, our method elaborately encode the densely sampled frame features into a compact yet discriminative representation.

5 Conclusions

We have proposed a novel compact video representation to detect video copies from large video collections. More and discriminative information is embedding to our final representation through sparse coded CNN features extracted from densely sampled video frames. Experimental results on the VCDB show that this presentation is advantageous over recent state-of-the-art CCD approaches. We significantly improve the recall rate about 10% with higher precision rate. In the future, we will investigate the effective indexing strategy for the proposed representation to fast and accurately retrieve very large scale video dataset.

Acknowledgements. This work is supported by National Natural Science Funds of China (61472059, 61428202).

References

1. Chou, C.L., Chen, H.T., Lee, S.Y.: Pattern-based near-duplicate video retrieval and localization on web-scale videos. IEEE Trans. Multimedia **17**(3), 382–395 (2015)
2. Coates, A., Ng, A.Y.: The importance of encoding versus training with sparse coding and vector quantization. In: Proceedings of the 28th International Conference on Machine Learning (ICML-2011), pp. 921–928 (2011)

3. Douze, M., Jégou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-filtering. IEEE Trans. Multimedia **12**(4), 257–266 (2010)
4. Douze, M., Jégou, H., Schmid, C., Pérez, P.: Compact video description for copy detection with precise temporal alignment. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 522–535. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15549-9_38
5. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recogn. **42**(3), 425–436 (2009)
6. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_24
7. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Dar-rell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
8. Jiang, Y.-G., Jiang, Y., Wang, J.: VCDB: a large-scale database for partial copy detection in videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 357–371. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_24
9. Jiang, Y.G., Wang, J.: Partial copy detection in videos: a benchmark and an evaluation of popular methods. IEEE Trans. Big Data **2**(1), 32–42 (2016)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region pro-posal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
13. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
15. Song, J., Yang, Y., Huang, Z., Shen, H.T., Hong, R.: Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: Proceedings of the 19th ACM International Conference on Multi-media, pp. 423–432. ACM (2011)
16. U.S. National Institute of Standards and Technology: Trec video retrieval evaluation. http://www-nlpir.nist.gov/projects/tv2011/#ccd
17. Tan, H.K., Ngo, C.W., Hong, R., Chua, T.S.: Scalable detection of partial near-duplicate videos by visual-temporal consistency. In: Proceedings of the 17th ACM International Conference on Multi-media, pp. 145–154. ACM (2009)
18. Tang, J., Hong, R., Yan, S., Chua, T.S., Qi, G.J., Jain, R.: Image annotation by k nn-sparse graph-based label propagation over noisily tagged web images. ACM Trans. Intell. Syst. Technol. (TIST) **2**(2), 14 (2011)
19. Tang, S., Zheng, Y.T., Wang, Y., Chua, T.S.: Sparse ensemble learning for concept detection. IEEE Trans. Multimedia **14**(1), 43–54 (2012)
20. Thomas, R.M., Sumesh, M.: A simple and robust colour based video copy detection on summarized videos. Procedia Comput. Sci. **46**, 1668–1675 (2015)

21. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theor. **53**(12), 4655–4666 (2007)
22. Wu, X., Hauptmann, A.G., Ngo, C.W.: Practical elimination of near-duplicates from web video search. In: Proceedings of the 15th ACM International Conference on Multi-media, pp. 218–227. ACM (2007)
23. Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative CNN video representation for event detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1798–1807 (2015)
24. Yang, Y., Zhang, H., Zhang, M., Shen, F., Li, X.: Visual coding in a semantic hierarchy. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 59–68. ACM (2015)
25. Zhao, W.L., Wu, X., Ngo, C.W.: On the annotation of web videos by efficient near-duplicate search. IEEE Trans. Multimedia **12**(5), 448–461 (2010)