This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

Partial Copy Detection in Videos: A Benchmark and An Evaluation of Popular Methods

Yu-Gang Jiang, Jiajun Wang

Abstract—The goal of partial video copy detection is to find one or more segments of a query video which have (transformed) copies in a large dataset. Previous related research in this field used either small-scale datasets or large datasets with simulated partial copies by imposing several pre-defined transformations (e.g., photometric changes) due to the extremely time-consuming annotation of real copies. It is still unknown how well the techniques developed on simulated datasets perform on real copies, which are much more challenging and too complex to be simulated. In this paper, we introduce a large-scale video copy database (VCDB) with over 100,000 videos, and more than 9,000 copy pairs found by manual annotation. A state-of-the-art system of video copy detection is evaluated on VCDB to show the limitations of existing techniques. We also evaluate deep learning features learned by two neural networks: one is independently trained on a different dataset and the other is tailored to deal with the copy detection task. Our evaluation suggests that all the existing techniques, including the deep learning features, are far from satisfactory in detecting complex real copies.

Index Terms—Video copy detection, benchmark dataset, frame matching, temporal alignment, deep learning.

1 Introduction

More and more videos are being transmitted online due to the prevalence of video capturing devices and network sharing platforms. Copyright infringement becomes an important problem because of the low cost of copying a video and distributing it on the Internet. Therefore, video copy detection, which concerns detecting copies in a large dataset automatically, has attracted significant research attention.

Due to the complex content variations on the Internet, such as scale and lighting changes, the task of video copy detection is very challenging. Local invariant features like the SIFT [1] and indexing structures such as the inverted file [2] have been popularly adopted for the purpose of precise and efficient copy detection. Despite the great progress, entire video-level copy detection, which requires a query video and a reference video to share very long copied segments, is the focus of many recent works [3], [4]. The datasets used in these works only have video-level annotations, i.e., whether two videos share copies of each other without the exact temporal location of copied segments. This greatly inhibits the research on the more fine-grained partial copy detection which involves finding one or more copied segments and their exact temporal location between a pair of videos. The need of copyright protection calls for more precise partial copy detection.

This work was supported in part by a National 863 Program (#2014AA015101) and a grant from the NSF China (#61572134). *Y.-G. Jiang and J. Wang are with the School of Computer Science, Fudan University, Shanghai, China (e-mail: {ygj,jiajunwang13}@fudan.edu.cn).*

Fig. 1. Three pairs of copy frames extracted from copy pairs in VCDB. All the copies were found from web videos through careful manual annotation. Different from existing datasets mostly generated "artificially" by imposing a few pre-defined transformations, the complex content variations in VCDB pose new challenges to video copy detection research.

The manual annotation of real copies is very hard and extremely time-consuming, so small scale datasets with *simulated* copies [2], produced by imposing a few pre-defined transformations, are used in recent researches. Despite the convenience of simulated datasets, it becomes an open question that how well the state-of-the-art approaches developed on simulated copies can be transferred to the application of real copy detection.

In order to address the aforementioned shortcomings of existing datasets, a large-scale video copy detection dataset (VCDB)[1] is elaborated in this paper. This dataset consists of more than 100,000 videos collected from the Internet, covering a wide range of

1. Available at: http://www.yugangjiang.info/research/VCDB/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

TABLE 1
Comparison of video copy detection datasets. VCDB is the only one containing real partial copies.

	Year	Partial Copy?	Type	Size of Dataset
Indyk et al. [5]	1999	N	Real	2,000 videos
Joly et al. [6]	2003	Y	Simulated	1,040 hours
Muscle-VCD [7]	2007	Y	Simulated	98 videos
CC_Web [3]	2007	N	Real	12,790 videos
TRECVID 2008 [8]	2008	Y	Simulated	200 hours
UQ_Video [4]	2011	N	Real	169,952 videos
VCDB	2014	Y	Real	100,528 videos

topics like movies and sports. Around 9,200 partial copies are found between about 6,000 pairs of videos by manual annotation. The video transformations in our dataset are very complex, some of which are shown in Figure 1.

In order to demonstrate the limitations of existing approaches, a popular method with state-of-the-art performance is evaluated on VCDB. In addition, we also conduct an evaluation of deep learning features, which have been popularly adopted in various image and video classification tasks but have rarely been used for video copy detection. Two types of deep learning features are used. One is based on the standard Convolutional Neural Networks (CNN) trained on a large set of images, while the other is generated by a network trained particularly to cope with content variations in the copy detection problem.

Our work has two main contributions. First, a large-scale dataset with real partial video copies is introduced, which requires significant efforts both in design and in annotation. Second, comprehensive benchmarking of existing approaches is conducted on our new dataset. The comprehensive benchmarking suggests that the systems with near-perfect results on the simulated datasets like TRECVID [9], [10] still have much room for improvement on our dataset. This calls for more research on detection algorithms of real partial video copies. This work is extended from a conference paper [11] with new proposals and evaluations of the deep learning features and expanded discussions throughout the paper.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Section 3 describes the construction and annotation of VCDB. Section 4 briefly introduces the baseline system with SIFT feature, Section 5 introduces two alternative deep learning features and Section 6 discusses the evaluation results. Finally, Section 7 concludes this paper.

2 Related Work

Datasets on video copy detection are first reviewed in this section. And then we review a few representative approaches and briefly discuss some popular deep learning methods.

2.1 Video Copy Detection Datasets

We summarize a few representative datasets in Table 1. Very few benchmark datasets on video copy detection have been released for cross-site comparison. Many researchers constructed datasets only for their own research. For instance, Indyk et al. [5] downloaded 2,000 clips of news, music videos and movie trailers with durations between 2 and 5 minutes. Copies were generated by imposing pre-defined transformations like inserting TV logos, camcording, changing frame rates, etc. A collection of 1,040 hours of TV videos are constructed by Joly et al. in [6]. The videos were stored in MPEG1 format, and included various contents such as commercials, news, sports and TV shows. Copies were also simulated in this dataset.

Law-To et al. [7] created the Muscle-VCD, which is perhaps the first well-known public benchmark dataset. The Muscle-VCD contains about 100 hours of videos from the Internet, TV archives and movies, which are stored in different resolutions and formats. Two kinds of queries are defined in the dataset: (1) ST1: entire video copy (normally between 5 minutes and 1 hour), where the videos may be slightly recoded and/or noised; (2) ST2: partial video copy, where two videos only share one or more short segments (between 1 second to 1 minute). The copied segments were simulated by imposing transformations, and were later used as queries to search for original versions.

The U.S. National Institute of Standards and Technology included a task on content-based copy detection in 2008 in its annual TRECVID evaluation [8]. A benchmark dataset was created and released to participants of this task every year. The 2008 version includes about 200 hours of TV programs and around 2,000 query clips, which has been used in many works [10], [2]. The queries were generated like Muscle-VCD by randomly sampling video segments from datasets and imposing transformations. This task was cancelled in 2011 due to near-perfect submitted results. However, the evaluations on our dataset will show that the near-perfect performance is not the situation for real partial copy detection.

Apart from the aforementioned simulated datasets, there are a few datasets containing real video copies collected from the Internet. Wu et al. [3] created

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

the popularly used CC_Web dataset, which contains 12,790 videos collected by searching videos on Google, YouTube and Yahoo!. The CC_Web was extended to UQ_Video dataset [4] by adding more background distraction videos. But these two datasets were created for the problem of near-duplicate video detection, which is different from copy video detection. Videos which are created from two sources but contain the same contents, such as the same scene captured by two cameras, are near-duplicate videos by definition, but not copy videos. In addition, the transformations existed between videos in the datasets are limited and easy to be detected; there are only video-level annotations without the exact timestamps of copies. These all make the datasets not suitable for benchmarking approaches of partial video copy detection.

2.2 Video Copy Detection Approaches

The works of copy detection systems can be divided as video-level copy detection and partial (sub-video-level) copy detection. Works dedicated to entire video-level copy detection used global features like color histogram and local features like the LBP [12], [3], [4]. The copies used in experiments were mostly with limited content variations, thus reasonably good results were reported.

The detection and localization of partial copies are more challenging, particularly under severe content variations, so more advanced techniques have been used. In [9], local features are extracted and quantized to the bag-of-visual-words (BoV) representations, which are then indexed by an inverted file structure for efficient search. In [2], local features are aggregated similar to the Fisher Vector [13], [14], which are indexed for efficient frame matching. Finally, a modified Hough voting scheme is used to produce the timestamps of copied segments. Similarly, bag-of-words representations of local features and inverted file indexing are also used in another system by Tan et al. [15]. But they additionally used a geometric consistency verification method to filter out outlier wrong matches of local features. In the end, the problem of finding the partial copies is formulated as a network flow optimization by constructing a temporal network using frame matching results.

As can be shown from the above works, most partial copy detection systems contain the following main parts: extraction of local features from frames, frame-level matching, and temporal alignment method to produce the final copied segments. These main parts have technical variations in different systems, such as the choice of the efficient descriptor matching method (e.g., using the product quantization [16] or its extended version [17]), the geometric verification scheme (e.g., using the Weak Geometric Consistency [9] or its variant [18]), or the final copy segment detection algorithm (e.g., using the Hough voting [2] or the temporal network [15]). The frame-level matching by local descriptors is almost the same as the techniques used in image-based object retrieval, which has been extensively studied in recent years [19], [9], [20], [21], [22], [23].

2.3 Deep Learning Features

In the past few years, deep learning has been extensively used in image and video analysis tasks with great success. The most noteworthy characteristic of deep learning features is that they are learned from training data by neural networks instead of hand-crafted. In [24], Krizhevsky et al. trained a deep convolutional neural network (CNN) on 1.2 million images in the ImageNet LSVRC-2010 contest to classify images into 1000 different classes. The resultant AlexNet network consists of 5 convolutional layers, 3 fully-connected layers with 60 million parameters and 650,000 neurons. In [25], features extracted from AlexNet are utilized to the problem of object detection benchmarking on PASCAL ROC dataset. A recent work by Fischer et al. [26] show that the local features learned by CNN are better than SIFT under most image variations.

Different from the typical CNN used for image classification, another deep neural network structure is Siamese network. Like "Siamese twins", siamese network has two identical sub-network to extract features from a pair of input data, and then links two features with a connection function. The structure of Siamese network makes it suitable to problems pertaining comparison. In [27], face patches were used by a Siamese network to train deep learning features for face identity prediction. Similar works [28], [29] also used Siamese network to train features for face verification with small variations on net structure. Despite the good performance of Siamese network on face problems, as far as we know, there are no works directly using a Siamese structure to train features for video copy detection.

As can be shown from above, for video copy detection, there are two prospective ways to embed deep learning features: to use a pretrained network such as Alexnet, or to train a new feature with siamese network from scratch. Training from scratch can incorporate video transformation knowledge into the features, which may help detect copies with strong content transformations.

3 Constructing VCDB

3.1 Video Collection

Collecting VCDB videos started from selecting video queries to be searched on two video-sharing websites: YouTube and MetaCafe. After careful selection, 28 queries were used, which cover a wide range of topics

Fig. 2. An example of a video pair containing multiple partial copies. Similar cases are abundant in VCDB.

such as commercials, movies, music videos, public speeches, sports, etc. Then we downloaded around 20 videos per query from the two video websites, which may contain partial copies. This gave us 528 videos in total (approximately 27 hours) to form a *core* dataset, leading to around 6,000 candidate pairs ($\binom{20}{2}\times 28$) to be annotated.

After that, an additional 100,000 videos were further collected to serve as background distraction videos. In order to avoid copies between videos in the core dataset and distraction videos, we skimmed over all these distraction videos. The core dataset and the distraction videos together form the final VCDB dataset.

3.2 Annotation

The annotation of 6,000 pairs of videos down to frame level is extremely hard and time-consuming, especially for video pairs that contain multiple short partial copies, as can be shown in Figure 2. We first planned to crowdsource this heavy task on websites like the Amazon MTurk. But to accurately locate the boundaries of copied segments is rather sophisticated and requires many precise operations, which is very hard to design a user-friendly interface for novice workers. So instead, seven part-time annotators were recruited and trained to perform the task.

In order to alleviate the hard work in annotation, we developed a carefully designed annotation tool. In the annotation tool, two videos can be viewed simultaneously at different timestamps to be compared, and the boundaries of found copies can be written into a database easily. To further reduce the effort, the transitivity property of copy relation is used in our annotation tool. Specifically, two segments which are copies to the same third segment tend to be copies of each other. Notice that the transitivity rule does not always hold, as the bridging segments may contain contents from two different sources that are copies to the two segments respectively (e.g., the transformation of picture-in-picture, as can be seen in the middle of Figure 1). Thus, the annotation tool will recommend *candidate* segment pairs by using the transitivity rule to be confirmed by annotators. The recommended segment pairs already have the boundaries specified, which greatly reduce the annotation task. In the end, the entire annotation work lasted about one month (around 700 man-hours).

3.3 Statistics

After the annotation, 9,236 pairs of partial copies were found in total. Figure 3 gives one example copy for each of the 28 queries. As can be shown, the transformations in VCDB are diverse and hard to be simulated by imposing a few pre-defined transformations, which makes VCDB superior to the existing datasets in benchmarking.

A few major transformations popularly used in simulated datasets are counted by running through all the 9,236 copy pairs. The final statistics show that about 36% of them contain "insertion of patterns", 18% are from "camcording", 27% have scale changes, and 2% are with "picture in picture" patterns. The simulated datasets usually have around the same amount of copies for each transformation, but the distribution of our VCDB is far from uniform, with "insertion of patterns" extremely high, and "picture in picture" patterns extremely low.

More statistics about VCDB are provided in Figure 4. We can see from Figure 4(a) that, among the video pairs that have at least one partial copy, nearly 80% of them contain only one copied segment, while as high as 20% contain multiple partial copies. Moreover, Figure 4(b) shows that short copies are very common in VCDB as 32% of the found copied segments are less than 10 seconds and another 28% are between 10 and 30 seconds. More importantly, according to Figure 4(c), 44% of the copies are shorter than $1/5$ of their parent videos and only 31% of them occupy over $4/5$ of the parent videos, which confirms the fact that most copies in VCDB are partial video segments.

4 A Baseline System

We implement a baseline system that has achieved very good performance on various datasets to demonstrate the capability of existing copy detection tech-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

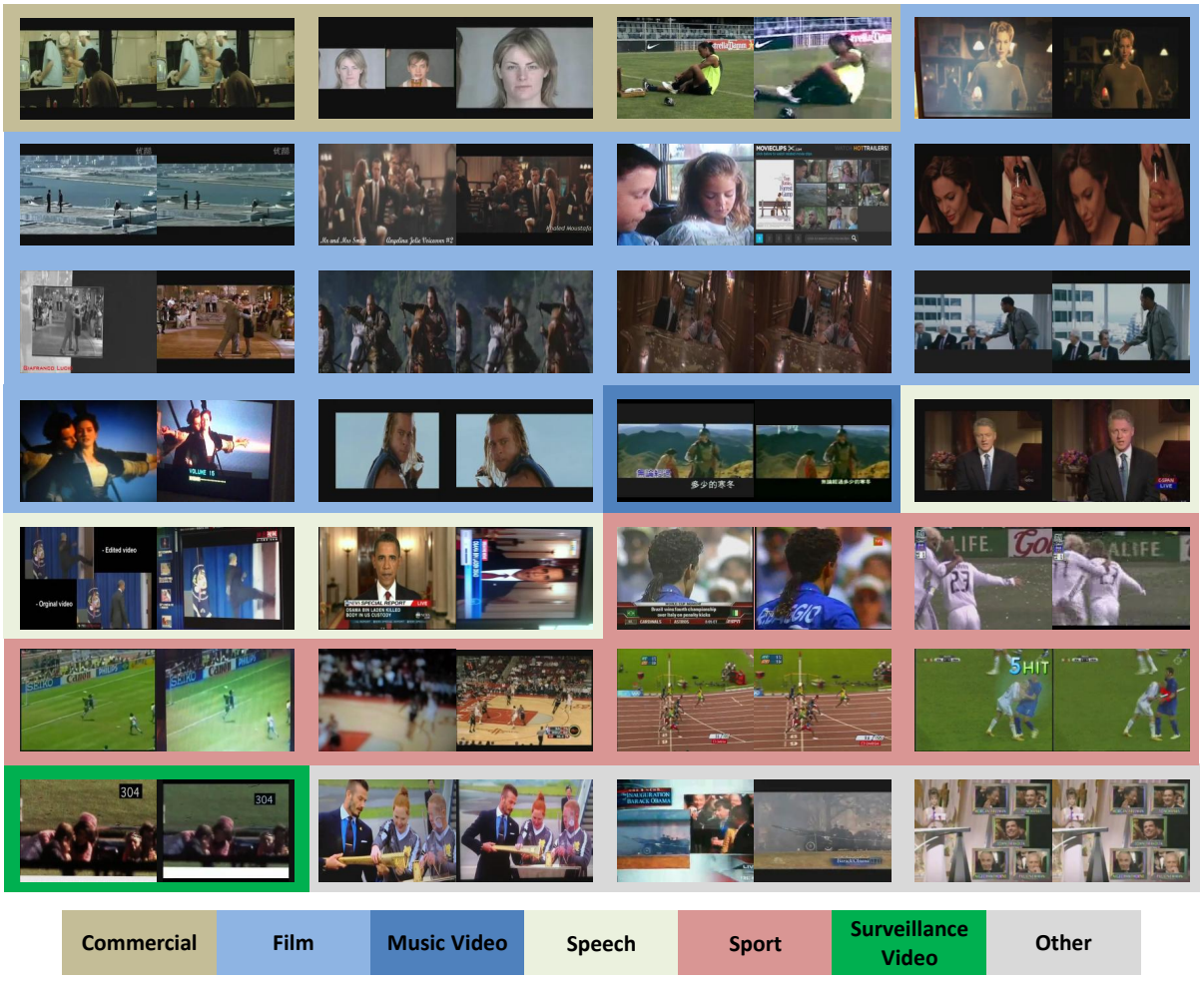


Fig. 3. Example copy frames from the videos downloaded by the 28 queries, ordered from left to right and top to bottom by topic categories (indicated by background color): commercials (3), films (11), music videos (1), speeches (3), sports (6), surveillance videos (1), and others (3). This figure is best viewed in color.

niques and to show the difficulty of our VCDB. Figure 5 shows the general framework of most state-of-the-art video copy detection systems, which is also the pipeline of our baseline system. For modules 1–4 of Figure 5, we use the techniques in the work of Jegou et al. [9]; for the remaining temporal alignment modules, two methods from [15] and [2] are adopted and compared. More details about the baseline system are described in the following sections.

4.1 Feature Extraction and Frame Representation

At first, frames are uniformly sampled from videos and local SIFT descriptors are computed and quantized by the standard BoV representation on each frame. The hierarchical k-means is then used to efficiently generate the codebook for BoV representation, which segments the SIFT feature space into many Voronoi cells.

In the experiments, we extract standard 128-dimensional SIFT features for each sampled frame using the VLFeat package[2]. Especially, the SIFT interest points are first detected based on local extrema of the Difference of Gaussian scale space, with each octave sampled at 3 intermediate scales, and then SIFT descriptors are computed on a 4×4 spatial histogram with bin size as 8. The hierarchical k-means contains 4 layer with branching factor as 15, leading to a codebook of $15^4 = 50625$ visual words.

4.2 Indexing and Hamming Embedding

We use an inverted file structure to index the quantized SIFT feature vectors for efficient frame matching. Additionally, Hamming embedding from [9] is incorporated in our system to reduce the quantization error of the standard BoV representation. The key idea of Hamming embedding is to further partition Voronoi cells into more subspaces by mapping each local descriptors to a binary code, with the similarity between descriptors being approximated by the Hamming distance between binary codes. Thus, two SIFT

2. http://www.vlfeat.org

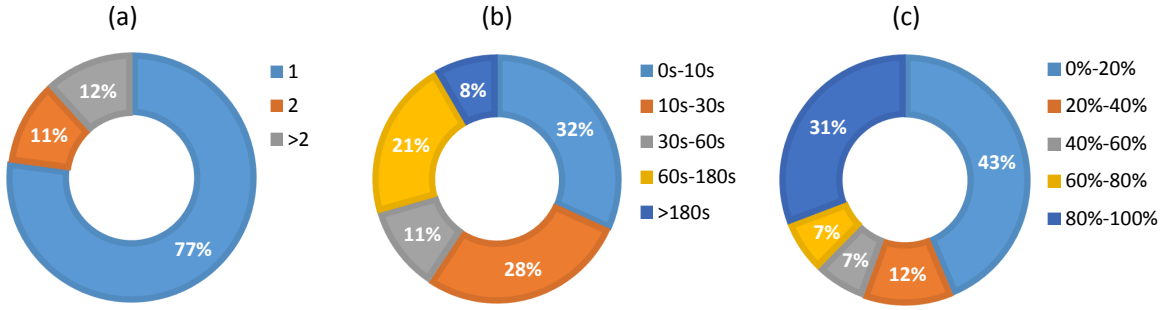


Fig. 4. Statistics of VCDB: (a) the number of partial copies per video pair, among those having at least one copy; (b) the duration of the partial copies; and (c) the percentage of the duration of the copy segments in the corresponding parent videos. From these statistics, we can see that as high as 20% of video pairs contain two or more partial copies; 32% of the found copied segments are less than 10 seconds and another 28% are between 10 and 30 seconds, which are very short; 44% of the copies are shorter than $1/5$ of their parent videos and only 31% of them occupy over $4/5$ of the parent videos. This confirms the fact that most copies in VCDB are partial video segments.

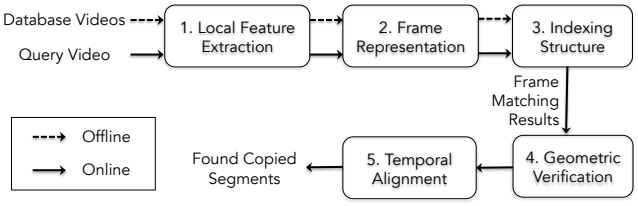


Fig. 5. The general framework of a video copy detection system.

descriptors are only matched when they are quantized to the same Voronoi cell and the Hamming distance is smaller than a threshold at the same time. This method is better than directly using a larger codebook (i.e., more Voronoi cells) since the latter will cause more quantization error [30].

4.3 Geometric Verification

Both the BoV representation and indexing structure with Hamming embedding ignore the geometric information like the orientation and scale of the SIFT descriptors, so the matching accuracy can be improved by using the technique of geometric verification which excludes "wrong" matches that are geometrically not consistent with the majority of matches. We use a weak geometric consistency (WGC) method from [9] to perform geometric verification. WGC adjusts the matching scores of video frames using the orientation and scale parameters computed in the SIFT descriptors. Specifically, the matching scores are enhanced if features are transformed by a consistent angle and scale, and reduced vice versa. Readers can get more details of Hamming embedding and WGC from [9].

4.4 Alignment by Temporal Network

The final step of video copy detection is to identify the copied segments by considering the frame-level matching results and the temporal information at the same time. By checking the temporal consistency between two videos, wrong frame matches can be further filtered out. In our system, two methods are adopted to perform the temporal alignment.

The first method is called temporal network [15], which formulate the problem as a network flow optimization process. Given a query video Q and a reference video R, the top-k similar frames to each frame in Q are collected by searching R. Then a directed temporal network is constructed by linking top-k frames chronologically according to their timestamps. The value (edge weight) of the link (edge) is the similarity score between the corresponding matched frames. Finally, the finding of copied segments is equivalent to searching for the longest path in the network under three constraints: the maximum temporal length between two successively aligned frames, the minimum temporal length of a copied segment, and the minimum similarity score between the matched frames.

4.5 Alignment by Temporal Hough Voting

The second method is called temporal Hough transform proposed in [2]. Denote $s(t_q, t_d) > 0$ as the matching score between a query frame at time t_q and a reference frame at time t_d. A histogram $h(\delta)$ is computed by accumulating the frame matching scores within a window of δ frames: $h(\delta) = \sum_{t_q \in Y} s(t_q, t_q + \delta)$, where Y is the set of timestamps of the query and $s(t_q, t_q + \delta) = 0$ if there are no matched reference frame with timestamp $t_q + \delta$. The final copied segments are produced by searching around peaks in the histogram. In practice, bursts of matches often exist among consecutive frames that are very similar, which causes the scores of the Hough histogram to be biased. So a re-weighting scheme is used to normalize the matching scores before computing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

histogram. A system [2] using this temporal alignment method achieved competitive results on the simulated TRECVID dataset.

4.6 Discussion

The guideline of choosing the above techniques in our baseline system is that they should be representative and have consistently good performance on multiple datasets, and the systems developed on methods in our system [9], [2] have achieved impressive performance on contests like the TRECVID [8]. The near-perfect results on simulated datasets maybe have sent a wrong signal that the problem of video copy detection is already solved, and therefore there were few systems developed in recent years. A recent system developed by Revaud et al. [17], which used VLAD [14] to aggregate the frame features and extended the product quantization [16] for event retrieval in large datasets, is perhaps the most related approach to our system. However, we found this pipeline produces slightly worse results than our baseline system. The reason of this is probably that the approach was designed for the problem of similar video event retrieval, which pays more attention on video semantics rather than visual patterns, and therefore cannot find copies after strong content variations. In the following section, we introduce a few new features for this challenging problem.

5 Deep Learning Features

Deep learning features, though being popularly used in many applications, have never been extensively tested in video copy detection tasks. In this work, we consider two types of deep learning features. Besides directly using a standard CNN pre-trained on a large image dataset, we also propose a Siamese convolutional neural network (SCNN) particularly trained for copy detection. The first CNN feature is used as a global frame-level feature, while the SCNN feature is learned from frame patches, and thus can be considered as a local feature. We discuss the detailed settings in the following sections.

5.1 Standard CNN

For each sampled video frame, a 4096-dimensional feature vector is extracted using the Caffe [31] implementation of the AlexNet in [24]. The video frames are directly rescaled to input size (i.e., 227×227) to feed in Alexnet. We use the output of the sixth fully connected layer as the frame-level feature vector. For details of the network architecture, please refer to [24]. After feature extraction, cosine similarity is used to measure the proximity of two video frames. Matched frames with similarities higher than a threshold are then further aligned with the temporal network approach described earlier.

Notice that various indexing methods may be used for fast frame comparison. In this work, we are more interested in knowing how the CNN feature performs on the copy detection task and how it compares with the traditional SIFT based methods. We leave indexing methods on this feature as a part of future work.

5.2 SCNN

Our SCNN has two identical sub-network (i.e., they share the same structure and the same weights), each of which contains 3 convolutional layer and 3 fully-connected layer, followed by a connection function layer and a loss layer. Figure 6 shows the detailed architecture of the SCNN. ReLU neuron [24] is used as activation function for each layer. The Euclidean distance is used as connection function to link the output of the last fully-connected layer.

Due to the twin structure of SCNN, copy and non-copy image patch pairs are used as training data. First, we randomly sample 64×64 patches from the video frames. Then, two random patches from different videos are paired as non-copy pair. For copy pairs, we apply four transformations on the original sampled patches: position translation, scaling, flipping, and optic change. The contrastive loss introduced in [32] is used as loss function in the training process. The goal of the loss function is to pull copy pair together and push non-copy pair apart. Caffe is used to train the network using Stochastic Gradient Descent (SGD) with mini-batches of size 256. Our training data includes 100,000 copy pairs, and the equal size of non-copy pairs. The copy pairs are randomly transformed in the aforementioned four ways. The learning rate is set 0.01 initially and decays as the training progresses. The momentum is set 0.9 constantly. As we do not witness overfitting as training in such manner, we train models until the loss does not decrease despite lowering learning rate.

Based on the experiences on using the CNN feature for image analysis, the best layer for feature extraction is an intermediate fully-connected layer, not the last layer, so we set the penultimate fully-connected layer as the feature layer. The neurons of the feature layer will be set different size to find the optimal feature dimension. Given an input frame patch, a sub-network of the SCNN can be used to generate the feature vector, which is then used as an alternative descriptor of the SIFT in the baseline system described in Section 4.

Concerning feature extraction, we first tried features by densely sampling hundreds of 64×64 patches from original video frames to feed in one sub-network of SCNN. However, we found this process tediously slow. Therefore we turn to a slightly different extracting pipeline. SCNN is split into two parts, namely convolutional part and fully-connected part. Each video frame is first rescaled to 256×256 to feed in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

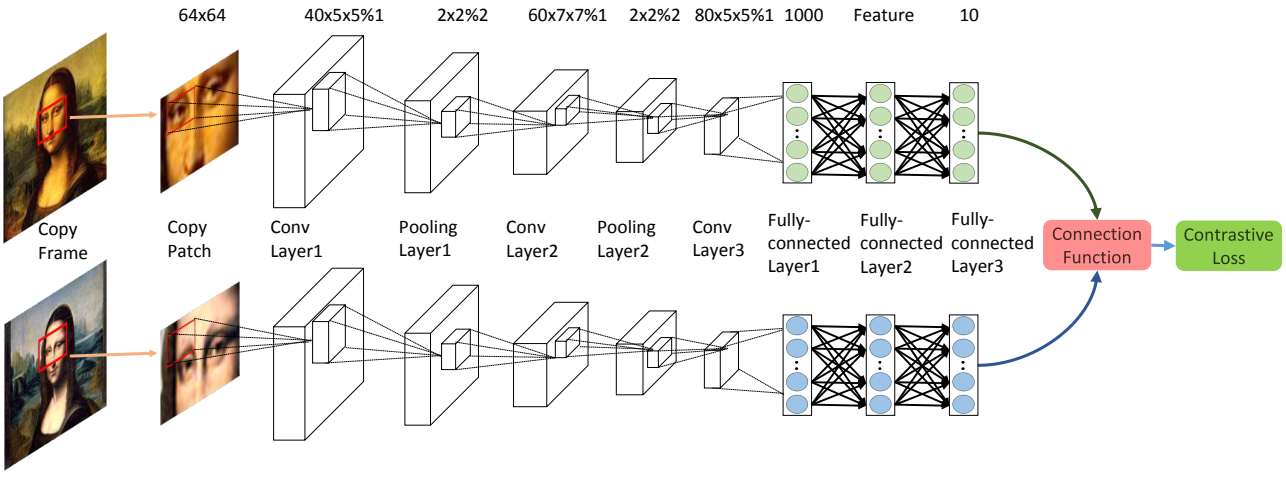


Fig. 6. An illustration of the architecture of our SCNN. For the convolutional layer, the dimension $40 \times 5 \times 5\%1$ means the layer has 40 feature maps with kernel size as 5×5 and stride as 1. Likewise, a $2 \times 2\%2$ pooling layer means its kernel size is 2×2 and its stride is 2. The second fully-connected layer is used for feature extraction in the testing phase.

the convolutional part, and then the final convolved output is densely sampled with stride as 2 to feed in the fully-connected part. In other words, we move the dense sampling process from the very beginning to after the final convolutional layer, saving the time of computation on convolution on overlapping areas of sampled frame patches. This extracting pipeline is nearly 20 times faster than the first one with nearly no performance loss.

After feature extraction, only the matching with the BoV representation and the temporal alignment steps are used afterwards. The step of geometric verification is dropped in the retrieval process as the SCNN descriptor has no scale and orientation information like the SIFT.

6 Experiments

In this section, we begin our experiments on a small and popular benchmark dataset, Muscle-VCD [7], to ensure that our baseline system is properly implemented. Then, thorough evaluations on VCDB are discussed.

6.1 Baseline System on Muscle-VCD

As described in Section 2.1, we focus on ST2 of partial copies to evaluate our baseline system. There are 21 queries in ST2, and the performance is measured by

$$QF = 1 - \frac{|missed\ frames|}{|groundtruth\ frames|}, and$$

$$QS = \frac{|correct| - |false\ alarm|}{|returned\ segments|},$$

the same as that in [7]. We uniformly sample two frames per second in all our evaluation. Even though sampling more frames may lead to slightly better results, the evaluation of this factor is beyond the focus of this paper.

Using temporal network as the alignment method in our baseline system, we achieve 0.81 for QS and 0.70 for QF. To our best knowledge, the highest results on this dataset are 0.86 and 0.76 respectively for the two measures, which were achieved by a similar method [15] to our system. The reason behind the small performance gap is that an enhanced version of WGC was used in [15], while our system only uses the standard WGC.

6.2 Baseline System on VCDB

The evaluation of the baseline system is presented in this subsection, in which we first report the performance on the core dataset, and then by incrementally adding more background distraction videos, experiments on large-scale dataset are conducted. We use all segments in 9,236 copy pairs as queries. The standard precision and recall are used as performance measure, which are commonly used in the evaluation of search systems. A correct detected copy pair requires that they both share frames to a ground-truth pair. We do not enforce a minimum number of shared frames because hitting a ground-truth pair with even one single frame will be useful in applications like copyright protection. More specifically, the segment-level precision (SP) and recall (SR) are defined as:

$$SP = \frac{|correctly\ retrieved\ segments|}{|all\ retrieved\ segments|}, and$$

$$SR = \frac{|correctly\ retrieved\ segments|}{|groundtruth\ copy|},$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

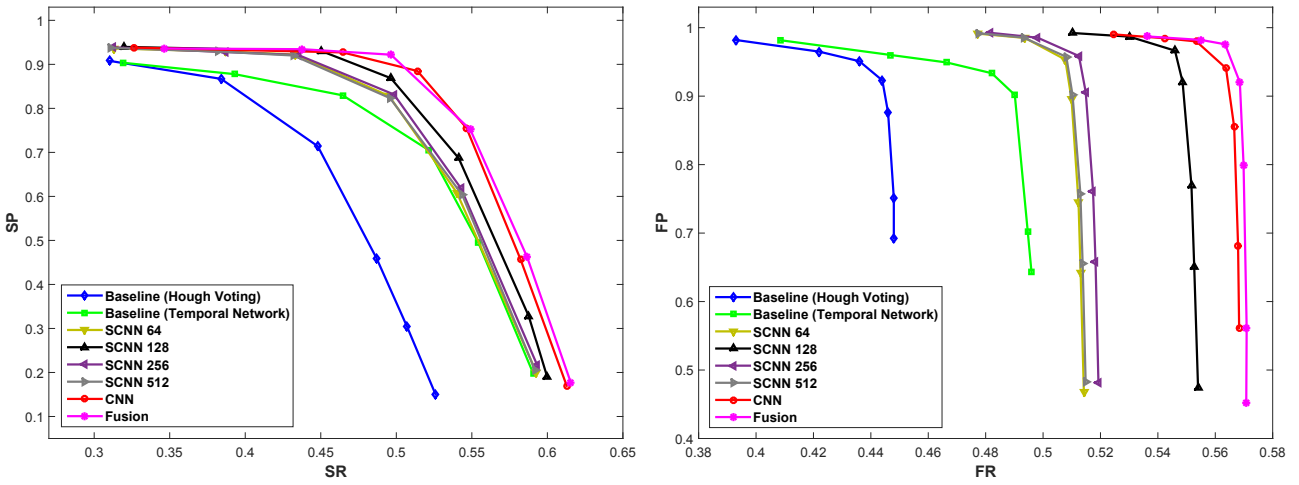


Fig. 7. Precision-recall curves of different methods on the core dataset of VCDB. The number following each SCNN method means the feature dimension (i.e., the number of neurons of the feature layer) of the network. The Fusion method means the fusion of the best three methods: Baseline (Temporal Network), SCNN 128, and CNN. **Left:** segment-level results. **Right:** frame-level results.

while the frame-level precision (FP) and recall (FR) are defined as:

$$FP = \frac{|correctly\ retrieved\ frames|}{|all\ retrieved\ frames|}, and$$

$$FR = \frac{|correctly\ retrieved\ frames|}{|groundtruth\ copy\ frames|}.$$

The frame-level measures are introduced as auxiliary criteria to show how accurate our baseline system is.

The results on the core dataset are shown as blue and green curves in Figure 7. We adjust the threshold of the minimum numbers of matched frames in alignment method to plot the precision-recall curves. The use of different alignment methods is the only technical difference behind the two curves. As can be shown, the temporal network method (the green curve) performs invariably better than Hough voting method (the blue curve), which is due to the effectiveness of explicitly enforcing several constraints in an optimization framework. Although temporal network runs slower than Hough voting, the difference is rather small since the number of matched frames is already limited after thresholding. The frame-level recall saturates at 0.5 and 0.45 for temporal network and Hough voting respectively, which shows that about half of copied frames can not be detected by our baseline system. The baseline system with temporal network method achieves very impressive results on Muscle-VCD, while on VCDB the segment-level recall is only 0.48 at a similar precision of 0.80. These all show that the performance on the core VCDB dataset is far from from satisfactory, which clearly verifies our argument that the partial copy detection is much more challenging under realistic situation.

Experiments on large-scale dataset is conducted by gradually adding more background distraction videos. As can be shown in Figure 8, we add 10,000, 30,000, 50,000, and 100,000 (the entire VCDB) videos sequentially, and the performance drops as more videos are added. Note that the degradation of performance is quite insignificant compared to the large number of videos added, particularly when precision is larger than 0.8. This shows that our system is not sensitive to background noises, which is very suitable for large scale real applications. Similar to results on the core dataset, more than 40% of partial copies are still not detected. These undetected copies indicate that the problem of partial copy detection has much room for improvement in the future research.

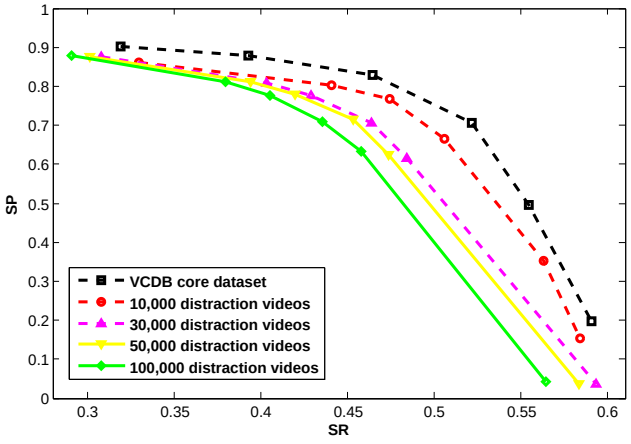


Fig. 8. Precision-recall curves of large-scale experiments on VCDB, using the temporal network method with different numbers of background distraction videos.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

TABLE 2
Segment-level and frame-level F-measure on the core dataset of VCDB.

	Baseline	CNN	SCNN	Fusion
Segment	0.5956	0.6503	0.6317	0.6454
Frame	0.6358	0.7101	0.6897	0.7143

TABLE 3
Time cost of issuing a 1-minute long video query to search in the core dataset of VCDB by only using one CPU thread.

	Baseline	CNN	SCNN
Time (s)	66	274	96

6.3 Deep Learning Features on VCDB

Finally, we evaluate the deep learning features on the VCDB core dataset. Results are also shown in Figure 7. After matching the similar video frames using both types of deep learning methods, the final copy results are generated by the temporal network method, since its performance is better than Hough voting. In order to find the optimal feature dimension, we set the feature layer of SCNN to different sizes and train a model from scratch separately for each size, which is shown in the Figure 7 as method "SCNN" followed by feature dimension.

As can be seen from the four lines of the SCNN method (all using triangular markers), the performance first increases and then decreases while the dimension becomes larger, and the optimal dimension falls on SCNN-128. This coincides with the fact that the commonly used SIFT feature is also 128-dimensional. Therefore, a dimension around 128 is suitable for a local image descriptor. We can also see from the figure that the deep learning features produce clearly better results than the SIFT-based baseline. These new features can retrieve about 2% more copy segments and about 5% more copy frames on the same precision value, which shows the superiority of trained deep features over hand-crafted features. In addition, even with the deep learning features, the frame-level recall still saturates at around 0.57, leaving a lot of undetected copies. Another interesting point observed from the results is that the CNN feature performs a bit better than the SCNN feature. This is because, in contrast to AlexNet CNN which is trained on 1.2 million images, the SCNN is trained on a very small set of samples. As the performance of the two approaches is fairly close, we expect that SCNN could possibly perform better once sufficient training data are provided. As the main purpose of this experiment is to compare the deep features with the hand-crafted SIFT, large scale experiment using these two features on the full VCDB is left for future work.

We also fuse Baseline, SCNN-128 and CNN together (indicated by "Fusion" in the figure), which produces slightly better results than CNN. Note that the fusion is done by simply merging the found copies by different methods, which may not be optimal. Figure 9 shows examples of a few failure cases, where we see strong content variations that cannot be tackled by the current approaches.

For the convenience of others to compare their algorithms on VCDB, we also report the *best* (highest) segment-level and frame-level F-measure (i.e., the harmonic mean of precision and recall) of different methods on the core dataset in Table 2, and the time cost of issuing a 1-minute long video query to search in the core dataset in Table 3. The time cost only includes the online part of processing a query, and is measured on an Intel Xeon E5-2690 3.00 GHz CPU by using only one thread without GPU intervention. The largest time cost is from the CNN method because it uses a brute-force manner to search similar video frames, while the other two use inverted-index. The SCNN method requires extracting densely sampled deep learning features, which makes it a lot slower than the baseline method.

6.4 Challenges of VCDB

We can see from the experiments above that the performance on VCDB is far from satisfactory. For applications of video copy detection like copyright protection, a high recall rate is very crucial, but the best segment-level and frame-level recall rate is only around 0.60. We think that the challenges of VCDB mainly comes from two reasons:

- **Complicated visual transformation**. The videos in VCDB are collected from the real world, so the pattern of copy transformations is very different from that of simulated copy datasets like TRECVID. Simulation has rules, while real videos are edited by people creatively, without clearly defined rules, which brings significant difficulty in detection. Besides, the competition task of TRECVID was to link transformed videos with their original untransformed version, there is hardly anyone knows which version is original in VCDB. The task with VCDB is to pair differently transformed videos, which is harder.
- **Complicated temporal structure**. For the TRECVID data, there are no multiple partial copies between a video pair, but in VCDB multiple partial copies are very common. Especially for movie videos, people tend to break down long movie videos into small clips and re-edit them into a new web video. This makes copies very short and numerous. When two of these fragmented videos come together, the temporal structure between them is complicated and very hard to detect.

Complicated visual transformation calls for more robust visual features, of which deep learning feature is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2016.2530714, IEEE Transactions on Big Data

Fig. 9. Six frame pairs that are not detected by our system, which contain very severe and complex content variations.

a promising choice. Complicated temporal structure reminds us to pay more attention to the temporal information embedded in videos. Maybe recurrent neural network can be used to collect temporal cues and encodes them into better features. Exploring the possibilities of deep learning on the problem of video copy detection is a recommended research direction, which certainly requires some additional efforts upon the recent progress of deep learning on image classification [33], [34] and video analysis [35], [36].

7 Conclusion

In this paper, a new dataset called VCDB is introduced, which is the only large scale dataset in the community containing real partial video copies. Near-perfect results have been attained on simulated datasets in recent research of video copy detection, which may have sent a wrong signal that this is a solved problem and thus has limited the progress in this field. We hope to reinvigorate this field by providing over 9,000 highly challenging partial copies and over 100,000 videos in VCDB.

Evaluations of a baseline system with the traditional SIFT feature and two deep learning features are conducted on VCDB. The far-from-perfect performance indicates that VCDB could arguably be a good benchmark for future research. Two temporal alignment methods are also evaluated with the result that the temporal network method is a better solution. The fact that the best recall rate on VCDB is only about 0.60 with an extremely low precision of 0.20 suggests that future research should pay more attention on the frame matching process to overcome the severe content variations in real situation. Moreover, two deep learning features perform much better than SIFT feature, which shows a promising research direction of deep learning, although significant efforts may be needed to train better networks tailored for the copy detection problem.

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] M. Douze, H. Jegou, C. Schmid, and P. Perez, "Compact video description for copy detection with precise temporal alignment," in *ECCV*, 2010.

[3] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *ACM MM*, 2007.

[4] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *ACM MM*, 2011.

[5] P. Indyk, G. Iyengar, and N. Shivakumar, "Finding pirated video sequences on the internet," in *Technical Report, Stanford University*, 1999.

[6] A. Joly, C. Frelicot, and O. Buisson, "Robust content-based video copy identification in a large reference database," in *CIVR*, 2003.

[7] J. Law-To, A. Joly, and N. Boujemaa, "Muscle-VCD-2007: a live benchmark for video copy detection," 2007, http://www-rocq.inria.fr/imedia/civr-bench/.

[8] U.S. National Institute of Standards and Technology, "TREC video retrieval evaluation," http://trecvid.nist.gov/.

[9] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008.

[10] M. Douze, H. Jegou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-ltering," *IEEE TMM*, vol. 12, no. 4, pp. 257–266, 2010.

[11] Y.-G. Jiang, Y. Jiang, and J. Wang, "Vcdb: A large-scale database for partial copy detection in videos," in *ECCV*, 2014.

[12] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *CIVR*, 2007.

[13] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007.

[14] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2007.

[15] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua, "Scalable detection of partial near-duplicate videos by visual-temporal consistency," in *ACM MM*, 2009.

[16] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE TPAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[17] J. Revaud, M. Douze, C. Schmid, and H. Jegou, "Event retrieval in large video collections with circulant temporal encoding," in *CVPR*, 2013.

[18] W. L. Zhao and C. W. Ngo, "Flip-invariant sift for copy and object detection," *IEEE TIP*, vol. 22, no. 3, pp. 980–991, 2013.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

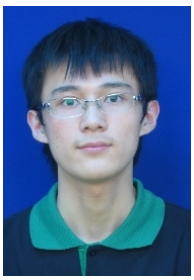
[20] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *CVPR*, 2010.

[21] R. Arandjelovi and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.

[22] Y. Avrithis and G. Tolias, "Hough pyramid matching:

Speeded-up geometry re-ranking for large scale image retrieval," *IJCV*, 2013.
[23] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," in *ICCV*, 2013.
[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
[26] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to sift," *arXiv preprint arXiv:1405.5769*, 2014.
[27] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *CVPR*, 2014.
[28] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *CVPR*, 2014.
[29] D. Yi, Z. Lei, and S. Z. Li, "Deep metric learning for practical person re-identification," *arXiv preprint arXiv:1407.4979*, 2014.
[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.
[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014.
[32] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.
[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.
[35] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, "Evaluating two-stream cnn for video classification," in *ICMR*, 2015.
[36] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *ACM MM*, 2015.

Yu-Gang Jiang received the Ph.D. degree in Computer Science from City University of Hong Kong, Kowloon, Hong Kong, in 2009. During 2008-2011, he was with the Department of Electrical Engineering, Columbia University, New York, NY. He is currently an Associate Professor of Computer Science at Fudan University, Shanghai, China. His research interests include computer vision and multimedia retrieval. He has authored more than 80 papers in these fields. He is an active participant of the annual NIST TRECVID Evaluation and has designed a few top-performing video retrieval systems over the years. He is an associate editor of Machine Vision and Applications and recently served as a program chair of ACM ICMR 2015. He is also one of the organizers of the annual THUMOS action recognition challenge. He work has led to many awards, including the 2014 ACM China Rising Star Award and the 2015 ACM SIGMM Rising Star Award.



Jiajun Wang received the B.Sc. degree in Physics from Fudan University, Shanghai, China, in 2013. He is currently working toward the M.Sc. degree with the School of Computer Science, Fudan University. His research interests include computer vision and multimedia retrieval.