SVD: A Large-Scale Short Video Dataset for Near-Duplicate Video Retrieval

Qing-Yuan Jiang†, Yi He‡, Gen Li‡, Jian Lin†, Lei Li‡ and Wu-Jun Li†
†National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, Nanjing, China
‡ByteDance AI Lab, Beijing, China
jiangqy@lamda.nju.edu.cn,{heyi,ligen.lab,lileilab}@bytedance.com,
linj@lamda.nju.edu.cn,liwujun@nju.edu.cn

Abstract

With the explosive growth of video data in real applications, near-duplicate video retrieval (NDVR) has become indispensable and challenging, especially for short videos. However, all existing NDVR datasets are introduced for long videos. Furthermore, most of them are small-scale and lack of diversity due to the high cost of collecting and labeling near-duplicate videos. In this paper, we introduce a large-scale short video dataset, called SVD, for the NDVR task. SVD contains over 500,000 short videos and over 30,000 labeled videos of near-duplicates. We use multiple video mining techniques to construct positive/negative pairs. Furthermore, we design temporal and spatial transformations to mimic user-attack behavior in real applications for constructing more difficult variants of SVD. Experiments show that existing state-of-the-art NDVR methods, including real-value based and hashing based methods, fail to achieve satisfactory performance on this challenging dataset. The release of SVD dataset will foster research and system engineering in the NDVR area. The SVD dataset is available at https://svdbase.github.io.

1. Introduction

Over the past decades, we have witnessed the explosive growth of video data in a variety of video sharing websites like YouTube[1], Instagram[2], and TikTok[3]. For example, 400 hours of new videos were uploaded to Youtube every minute and one billion hours of content was watched on YouTube every day in February 2017[4]. With billions of videos being available on the internet, it becomes a major challenge to perform near-duplicate video retrieval (NDVR) from a large-scale video database. NDVR aims to retrieve the near-duplicate videos from a massive video database, where near-duplicate videos are defined as videos that are visually close to the original videos [32]. For example, the videos might be slightly modified by the users to bypass the detection, and the modified videos can be treated as near-duplicate videos of the original videos. These modifications can be caption insertion, border insertion and so on. An NDVR system has been a necessity on content platforms with many applications, including video recommendation, video search, and copyright infringement detection. Hence, NDVR has become a hot research topic, and there have appeared a lot of methods for NDVR [32, 10, 8, 4, 33, 29, 1, 24, 16, 18, 2, 23, 13, 30, 19, 6].

Existing NDVR methods can be divided as video-level methods and frame-level methods. Video-level methods, including layer-wise convolutional neural network (CNNL) [12], vector-wise convolutional neural network (CNNV) [12] and deep metric learning (DML) [13], try to represent each video as a global feature. Frame-level methods, including spatio-temporal post-filtering [4], circulant temporal encoding (CTE) [24] and temporal matching kernel (TMK) [23], extract features for each frame of the video. In the meantime, to advance the research of NDVR, several video datasets have been introduced in recent years, including CCWEB [32], UQ_VIDEO [29], VCDB [9], MUSCLE_VCD [14], TRECVID [22] and so on. However, all of them are for long videos with average duration longer than 60 seconds.

In recent years, short videos with duration less than 60 seconds have become increasingly popular on social media platforms. Users have strong incentive to copy a hot short video and upload a modified version on these platforms to gain attention. With the increasing in short video data, there appear new difficulties and challenges for detecting near-duplicate short videos. Some of the new difficulties and challenges are listed as follows. Firstly, most long videos are generated by professional photographers with

[1]https://www.youtube.com
[2]https://www.instagram.com
[3]https://www.tiktok.com
[4]https://en.wikipedia.org/wiki/YouTube

cameras, while most short videos are generated by amateurs with mobile devices. Hence, the short videos might contain some new types of near-duplicates, e.g., horizontal/vertical screen videos and camera shaking videos. Secondly, as the cost of editing a short video is cheaper, users might prefer to edit a short video. Hence, the number of near-duplicate short videos is larger than that of near-duplicate long videos. Therefore, there is an urgent need of a large-scale short video dataset for NDVR task.

In this paper, we introduce a new large-scale short video dataset, called SVD, to foster research of NDVR for short videos. The main contributions of this paper are listed as follows:

- The introduced SVD dataset contains over 500,000 short videos and over 30,000 labeled videos for NDVR task. To the best of our knowledge, SVD is the first large-scale short video dataset for NDVR task. Compared with existing NDVR datasets, SVD dataset is the largest one.
- With hard labeled positive/negative videos mined by multiple strategies, SVD dataset is challenging for NDVR. Furthermore, we design some temporal and spatial transformations to mimic user behavior in real applications and construct more difficult and challenging variants of SVD.
- We perform two categories of retrieval to evaluate the performance of existing state-of-the-art NDVR methods on SVD dataset, i.e., real-value based retrieval and hashing based retrieval. Experiments demonstrate that these NDVR methods cannot achieve satisfactory retrieval performance on SVD dataset. Hence, the release of SVD dataset will foster the research of the NDVR area.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. In Section 3, we describe the dataset collection strategies in detail. In Section 4, we introduce some temporal and spatial transformations applied to SVD dataset. In Section 5, we carry out experiments on SVD dataset. At last, we conclude our paper in Section 6.

2. Related Work

We briefly review the datasets for NDVR task in this section. Specifically, related datasets include CCWEB [32], UQ_VIDEO [29], VCDB [9], MUSCLE_VCD [14], and TRECVID [22] datasets.

CCWEB [32] dataset contains 24 query videos and 12,790 labeled videos. The authors utilize 24 text queries, e.g., "The lion sleeps tonight" and "Evolution of dance", to retrieve the videos from Youtube, Google Video, and Yahoo! Video. The returned videos contain 27% redundant videos. Then the authors collect 12,790 videos as labeled set. The average duration for this dataset is 151.02 seconds. In this dataset, over half of the queries are about dancing and singing, which is lack of diversity.

UQ_VIDEO [29] is an extended dataset of CCWEB. The authors utilize 24 query videos and 12,790 labeled videos of CCWEB as the query set and labeled set for UQ_VIDEO dataset, respectively. Then the authors construct a background distraction set with 119,833 videos. The videos in background distraction set are usually treated as negative, but the labels are not verified by humans. In the end, the authors collect 132,647 videos in total. Although UQ_VIDEO is larger than CCWEB, it is also lack of diversity due to the limited number of queries. Furthermore, for all background distraction videos, this dataset only provides HSV [26] features and LBP [7] features of all key frames, and the original videos are not publically available.

VCDB [9] dataset utilizes the same 528 videos to construct both query set and labeled set. Furthermore, the authors provide 100,000 background distraction videos. Thus this dataset contains 100,528 videos in total. Furthermore, VCDB dataset is originally proposed for copyright detection task, and only provides 9,236 copied segment labels. However, for NDVR task, we need video-level pairwise labels to denote whether a candidate video is the near-duplicate video of the query video or not. Hence, we filter redundant copied segment pairwise labels and get 6,139 video-level pairwise labels for NDVR task. Please note that all 6,139 video-level pairwise labels are positive. The average duration of the VCDB dataset is 72.77 seconds.

MUSCLE_VCD [14] collects 18 videos to construct query set. Then the authors utilize query videos to generate 101 videos as labeled set based on some predefined transformations. Thus MUSCLE_VCD dataset collects 119 videos in total.

TRECVID [22] dataset utilizes 11,256 query videos to construct query set. Then the authors use query videos to generate 11,503 videos as labeled set based on some predefined transformations. Thus TRECVID dataset collects 22,759 videos in total.

The above datasets have been widely used for NDVR task. All of these datasets are long video datasets and have different shortcomings. Specifically, the videos of TRECVID and UQ_VIDEO datasets are not publicly available. MUSCLE_VCD and TRECVID datasets are small-scale and the labeled videos of these two datasets are generated by the authors of the datasets rather than the users of real video platforms. CCWEB and UQ_VIDEO datasets are lack of diversity. VCDB dataset only contains positive pairwise labels. The second to the sixth columns of Table 1 list the statistics of the aforementioned datasets. From Table 1, we can find that all existing NDVR datasets are long videos with average duration longer than 60 seconds.

Table 1. Comparison between SVD and existing datasets. As the original videos in background distraction set of UQ_VIDEO are not publicly available and we cannot access MUSCLE_VCD and TRECVID datasets, some statistics of these three datasets are N/A.

Item	CCWEB	UQ_VIDEO	VCDB	MUSCLE_VCD	TRECVID	SVD
#query videos	24	24	528	18	11,256	1,206
#labeled videos	12,790	12,790	528	101	11,503	34,020
#positive pairs	3,481	3,481	6,139	N/A	N/A	10,211
#negative pairs	9,311	9,311	0	N/A	N/A	26,927
#background distraction videos	0	119,833	100,000	0	0	0
#probable negative unlabeled videos	0	0	0	0	0	526,787
#total videos	12,814	132,647	100,528	119	22,759	562,013
Average duration (in second)	151.02	N/A	72.77	3,564.36	131.44	17.33
Total duration (in hour)	539.95	N/A	2027.60	100	420	2704.96
Video publically available	√	×	√	√	×	√

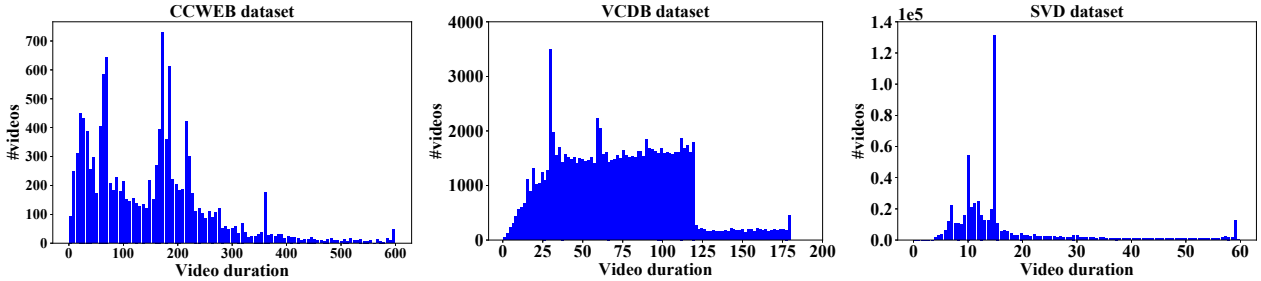

Figure 1. Video duration comparison on CCWEB, VCDB and SVD datasets. Note the average duration of our constructed SVD is significantly shorter than that of CCWEB and VCDB.

3. SVD: A Large-Scale Short Video Dataset

In this section, we describe the dataset collection strategies for constructing our large-scale short video dataset called SVD.

All videos in SVD dataset are crawled from a large video website Douyin[5] and the video format is ".mp4". The duration of most videos is less than 60 seconds. We crawled an ambient set containing over 100 million short videos, from which we select videos and construct SVD. The SVD dataset is divided into three subsets, i.e., the query set, the labeled set and the probable negative unlabeled set. First, we collect 1,206 videos as the query set. Then we utilize multiple strategies to mine hard positive/negative candidate videos for annotation. Unlike the candidate videos which are randomly crawled in existing datasets, the candidate videos in SVD are hard by using multiple strategies for selection. Hence, we call these candidate videos as *hard* positive/negative candidate videos. After human annotation, we collect 34,020 labeled videos to get the labeled set, which includes 10,211/26,927 labeled positive/negative video pairs. Besides this, by utilizing a pairwise similarity filtering strategy, we collect 526,787 videos as probable negative unlabeled set rather than background distraction set. Here, the videos in probable negative unlabeled set are the negative videos which aren't verified by humans. Unlike background distraction videos which are crawled randomly in UQ_VIDEO and VCDB datasets, we utilize a filtering strategy to ensure that the videos in the probable negative unlabeled set are not near-duplicate videos of the query videos with high probability. Hence, the videos in probable negative unlabeled set are more suitable to be treated as negative than those in background distraction set. In the last column of Table 1, we present the statistics about SVD dataset. From Table 1, we can find that the average duration of the SVD dataset is only 17.33 seconds, which is shorter than other datasets. Furthermore, SVD is the largest dataset among all datasets in Table 1. In Figure 1, we further illustrate the distribution of durations for CCWEB, VCDB and SVD datasets. From Figure 1, we can see that most of the videos are short in SVD dataset compared with CCWEB and VCDB. In the rest of this section, we will describe the detailed construction strategies.

[5]http://www.douyin.com

3.1. Query Set

We crawl 1,206 videos, each with more than 30,000 "likes", as the query set. All of these queries were uploaded in November 2018. To ensure diversity, the contents and types of these query videos are made as diverse as possible. Specifically, the video contents of the query videos contain portrait, landscape, game video, animation and so on. The query videos also contain a variety of video types including vertical screen video, horizontal screen video and so on. Figure 2 illustrates some randomly sampled query videos.

Figure 2. Example of query videos in SVD. Each block represents a video with multiple frames.

3.2. Labeled Set

To construct the labeled set, we first choose some videos as candidate videos for annotation. All candidate videos are divided into positive (near-duplicate) candidate videos and negative candidate videos, which respectively denote the videos we expect to be annotated (labeled) as positive and negative videos of the corresponding query videos.

To mine hard positive/negative candidate videos for annotation, we utilize multiple strategies to select candidate videos from the ambient set. The strategies include iterative retrieval, transformed retrieval, and feature based mining. Among these strategies, the first two strategies are mainly used for mining hard positive candidate videos and the last strategy is used for mining hard negative candidate videos.

We collect nearly 50,000 video pairs for annotation. These video pairs are labeled by human annotators. Annotation costs over 800 hours in total. After removing the videos inappropriate for public release, we collect 1,206 queries and 34,020 labeled videos. In the rest of this subsection, we will describe the details of the three strategies for selecting candidate videos.

Iterative Retrieval To mine hard positive candidate videos, we utilize an interactive retrieval method to annotate the positive candidate videos. This method can be divided into the following three steps. Firstly, for a given query video, it retrieves through the ambient set to get the candidates by using a variety of methods, including LBP [21] and BSIFT [35] feature based retrieval methods. Secondly, human annotators label these candidates for each query and select the positive ones. Lastly, the selected positive videos are further fed into the first step to retrieve more positive candidates. The whole process is repeated for several times until no more positive videos can be found for a given query.

Because the interactive retrieval procedure requires low latency, we only employ LBP [21] and BSIFT [35] features during this procedure. More advanced features and similarity calculation methods are utilized for the following transformed retrieval procedure.

Transformed Retrieval We also apply various transformations, such as rotation and cropping, on query videos to get transformed videos. And then we use the transformed videos as queries to search over the ambient set. Specifically, we utilize LBP, BSIFT, and deep features based retrieval methods to select the candidate videos. Then we select the top-5 to top-10 results as candidate videos for further human annotation.

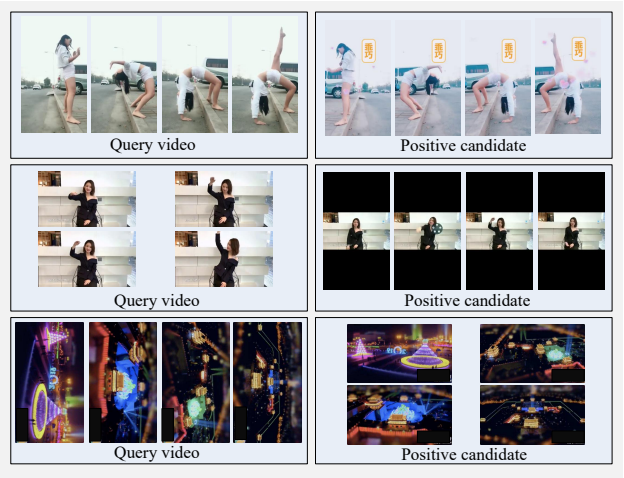

Figure 3. Example of hard positive candidate videos. Top row: side mirrored, color-filtered, and watermasked. Middle row: horizontal screen changed to vertical screen with large black margins. Bottom row: rotated.

In Figure 3, we show some query videos and their hard positive candidate videos mined by interactive retrieval and transformed retrieval. In Figure 3, the candidate videos are near-duplicate videos by various transformations including mirror transformation, color-filtered transformation, black border insertion, and rotation transformation.

Feature based Mining To mine hard negative candidate videos, we select 30,000 videos as candidate videos from the ambient set which were uploaded from June 2018 to August 2018. As the uploading dates of these candidate videos are earlier than those of the videos in our query set, we can expect that most candidate videos are not near-duplicate videos of the query videos. We extract different types of features to calculate the similarity between candidates and query videos. The features include hand-crafted features (LBP and BSIFT) and deep features. For each

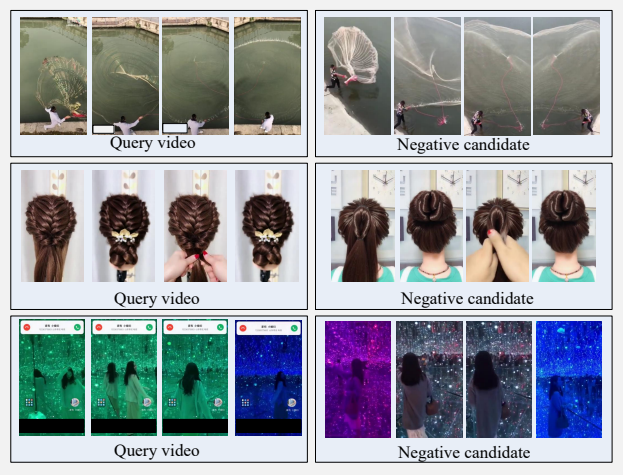

Figure 4. Example of hard negative videos. All the candidates are visually similar to the query but not near-duplicates.

query video, we select the top-5 to top-10 similar videos as candidate videos for human annotation.

Figure 4 illustrates some examples of query videos and the corresponding negative candidate videos, where the candidate videos are mined based on deep features. In the example at the top row, a man is casting a net into the water. In the example at the middle row, a girl is doing her hairstyle in a barbershop. In the example at the bottom row, a girl is playing in a room decorated with illuminations. However, as the persons in each video pair are different, all of these video pairs are not near-duplicate videos although they are very similar.

3.3. Probable Negative Unlabeled Set

We first select a subset of 700,000 videos from the ambient set as candidates for probable negative unlabeled videos, which are defined as negative videos without human annotation. After extracting a variety of frame and video features, we calculate the pairwise similarity between query videos and the candidate videos. The candidate videos which might be the near-duplicate videos of query videos with high probability will be filtered. Then the remaining candidate videos are selected as probable negative unlabeled videos. Specifically, we utilize BSIFT features and aggregated deep features to calculate similarity between query videos and candidate videos. The BSIFT features are used to calculate the Jaccard similarity, and only those videos whose similarities to *all* queries are 0 can be selected as candidate videos. Then the aggregated deep features are used to calculate video-level similarity based on Euclidean distance, and we further filter about 5% videos which have the smallest similarities to *all* queries. In the end, we obtain 526,787 videos for the probable negative unlabeled set.

To verify that the videos obtained by the above procedure are truly probable negative, we randomly sample 100 videos from the probable negative unlabeled set and invite human annotators to label them against each of the query videos. None of these videos is labeled as near-duplicate of the queries. Therefore, the videos in the probable negative unlabeled set are not near-duplicates of the query videos with high probability.

4. Transformations

In real applications, users might prefer to copy hot videos to gain attention. At the same time, these users usually choose to modify their copied videos slightly to bypass the detection. These modifications contain video cropping, border insertion and so on.

To mimic such user behavior, we define one temporal transformation, i.e., video speeding, and three spatial transformations, i.e., video cropping, black border insertion, and video rotation. Specifically, the video speeding transformation contains video speeding up and speeding down. This type of transformation is designed to simulate video acceleration or deceleration. In real applications, users might crop the videos to zoom in or out the original videos, which can be performed by frame cropping. Furthermore, users might insert borders, like black borders, to fit different video size. In addition, there exist many mobile-phone videos which are taken horizontally or vertically. When users upload these videos, they might rotate their videos.

These transformations are widely applied in the video re-creation procedure. By performing these transformations, harder candidates can be generated and we can construct more challenging datasets. Please note that the above transformations are used as illustrating examples, and users can define their own transformations based on their needs.

5. Experiments

We perform experiments to study the retrieval performance on SVD dataset and other NDVR datasets. We adopt two categories of NDVR methods, i.e., real-value based NDVR methods and hashing based NDVR methods. In real applications, real-value based NDVR methods usually suffer from high storage cost and low query speed. To avoid high storage cost and enable fast query speed, hashing based methods [3, 31, 34, 29, 11, 27, 6] have also been adopted for NDVR.

5.1. Datasets

As TRECVID and MUSCLE_VCD are too small and the original videos in background distraction set are not available for UQ_VIDEO, we select CCWEB [32] and VCDB [9] for comparison with SVD. We adopt four transformations defined in Section 4 to construct more challenging variants of SVD. Specifi-

cally, we utilize $\text{SVD}_{transformation}$ to denote a variant of the SVD dataset, where the labeled positive videos are *replaced* by the corresponding transformed videos. Here the $transformation$ denotes the transformations defined in Section 4, i.e., $transformation \in \{Cropping, BlackBorder, Rotation, Speeding\}$. Please note that we adopt acceleration transformation for $\text{SVD}_{Speeding}$. For all datasets, the groundtruth videos of a given query video are defined as the labeled positive videos.

5.2. Benchmark and Evaluation Protocol

5.2.1 Benchmark

For real-value based methods, we adopt four widely used real-valued NDVR methods, including three video-level methods, i.e., layer-wise convolutional neural network (CNNL) [12], vector-wise convolutional neural network (CNNV) [12] and deep metric learning (DML) [13], and one frame-level method, i.e., circulant temporal encoding (CTE) [24].

In real applications, real-value based methods might be impractical for massive videos. Hence, we also adopt some hashing methods for evaluation. Specifically, we adopt four hashing methods, including one data-independent method, i.e., locality sensitive hashing (LSH) [3], two unsupervised hashing methods, i.e., iterative quantization (ITQ) [5] and isotropic hashing (IsoH) [11], and one supervised hashing method, i.e., Hamming distance metric learning (HDML) [20], for evaluation. In this paper, we just use four hashing methods for demonstration, although more sophisticated hashing methods can be adopted to further improve the performance [15].

For real-value based NDVR methods, following the setting of DML [13], we utilize VGG16-Net [28] pre-trained on ImageNet [25] to extract 4096D deep features for every frame. For all datasets, we set $fps = 1$ for fair comparison[6]. After extracting deep features for each frame, we utilize the same normalization strategy as that in DML, i.e., zero-mean and L_2-normalization, to generate video-level deep features. DML is a triplet-based deep metric learning method. For all datasets, we utilize *hard triplets* sampling strategy proposed by [13]. For CNNL and CNNV methods, we also utilize 4096D deep features extracted by VGG16-Net pre-trained on ImageNet. For all datasets, we randomly sample 50,000 frames to learn 300 centers by k-means algorithm for CNNL and CNNV methods. For hashing based methods, we also use the 4096D deep features extracted by VGG16-Net to perform hashing learning for fair comparison. For all baselines except CNNL, CNNV and CTE, source code is kindly provided by their authors. For CNNL, CNNV and CTE, we carefully implement these methods.

[6]CTE achieves higher accuracy with $fps = 15$ on CCWEB dataset. In this paper, we set $fps = 1$ for fair comparison.

For real-value based NDVR methods, Euclidean distance is used to rank the retrieved data points. For hashing based NDVR methods, we learn a binary code for each video. Then the Hamming distance is used as the metric to rank the retrieved data points.

To further improve the retrieval accuracy for hashing methods, we can utilize reranking strategy. Specifically, we first use Hamming distance to generate a ranked list for all returned videos. Then we select top-N returned videos to run reranking algorithm. During the reranking procedure, we calculate Euclidean distance between query video and the selected top-N videos based on deep features and get the final ranked list for the selected N videos based on the Euclidean distance.

5.2.2 Evaluation Protocol

For CCWEB and VCDB datasets, following the setting of DML [13], we utilize the query set and labeled set as training set. During testing procedure, we utilize the query set as test set and the labeled set as database for CCWEB dataset. Then the retrieval procedure is performed by adopting test set to retrieve database. For VCDB dataset, we select query set as test set. Furthermore, we utilize the labeled set and background distraction set as database. For SVD dataset, we randomly select 1,000 query videos from query set and their labeled videos as training set. During testing procedure, we utilize the remaining 206 query videos from query set as test set. Furthermore, the corresponding labeled set and the whole probable negative unlabeled set are utilized as database.

We utilize mean average precision (MAP) and top-K MAP as evaluation metrics. Specifically, for each query video $\mathbf{v}_q$, the average precision (AP) is calculated according to the following equation:

$$AP(\mathbf{v}_q) = \frac{1}{R_q} \sum_{k=1}^{M} P_q(k) \mathbb{1}_k, \tag{1}$$

where R_q is the number of labeled positive videos, M denotes the number of videos in the database, $P_q(k)$ is the precision at cut-off k in the ranked list for video $\mathbf{v}_q$ and $\mathbb{1}_k$ is an indicator function which equals 1 if the k-th returned video is the groundtruth of query video, otherwise $\mathbb{1}_k = 0$. Then given n query videos, we can calculate MAP as follows:

$$MAP = \frac{1}{n} \sum_{q=1}^{n} AP(\mathbf{v}_q).$$

The top-K MAP can be calculated similarly by setting $M = K$ in Equation (1). Furthermore, we also compare the storage cost and retrieval time for real-value based NDVR methods and hashing based NDVR methods.

Table 2. MAP (%) for real-value based NDVR methods.

Method	CCWEB	VCDB	SVD	$\text{SVD}_{Cropping}$	$\text{SVD}_{BlackBorder}$	$\text{SVD}_{Rotation}$	$\text{SVD}_{Speeding}$
DML	97.01	78.98	78.47	54.07	68.17	15.59	76.70
CNNL	95.47	49.87	55.55	15.61	18.63	0.15	51.80
CNNV	95.60	45.19	19.09	6.31	6.94	0.22	15.45
CTE	90.08	41.42	50.97	16.48	32.66	2.84	16.23

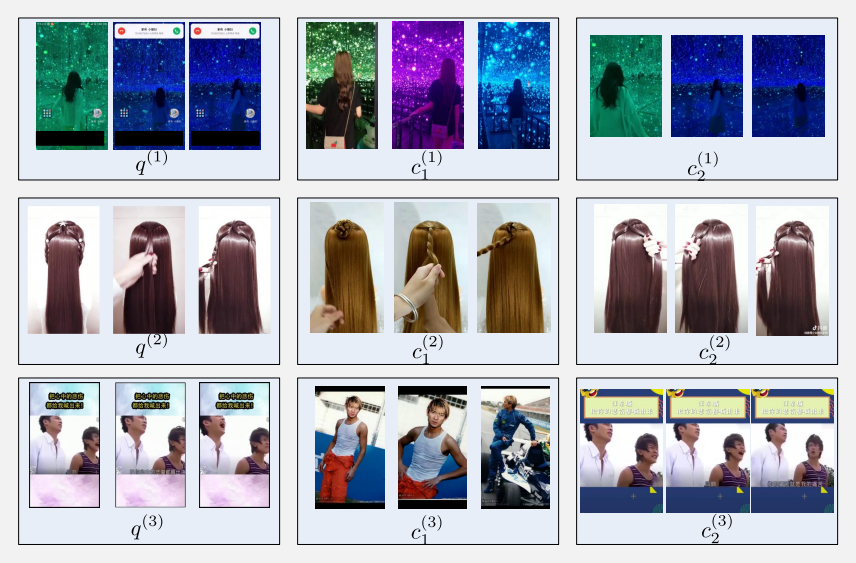

Figure 5. Bad cases of DML method.

5.3. Real-Value based NDVR

Accuracy We report MAP for DML, CNNL, CNNV, and CTE on CCWEB, VCDB and SVD datasets in Table 2. From Table 2, we can find that on CCWEB dataset, DML, CNNL, and CNNV methods can achieve similar promising retrieval accuracy. Furthermore, we can also find that the retrieval accuracy on SVD is far from satisfactory, which is similar to the phenomenon on VCDB dataset.

The MAP results on $\text{SVD}_{transformation}$ datasets are also presented in Table 2. From Table 2, we can see that for all transformations, the accuracy will be deteriorated, especially for spatial transformations.

Bad Case Analysis We present some bad cases of the best baseline (DML) on SVD dataset in Figure 5. In Figure 5, each element is a video which is shown as three representative frames. Each row contains a query video $q^{(i)}$ and its first returned video $c_1^{(i)}$ and second returned video $c_2^{(i)}$ according to the ranked list of DML. In all cases, the first returned video $c_1^{(i)}$ isn't the groundtruth of $q^{(i)}$ but the second returned video $c_2^{(i)}$ is the groundtruth of $q^{(i)}$.

For the first example shown in the first row, the query video $q^{(1)}$ and its groundtruth video $c_2^{(1)}$ show that a girl is walking in a room which is decorated with illuminations. Compared with $c_2^{(1)}$, the query video $q^{(1)}$ might be a video recorded by a smart-phone and its groundtruth video $c_2^{(1)}$ is edited by cropping. The video $c_1^{(1)}$ shows that another girl in a black T-shirt is walking in a room which is very similar to the room of query video $q^{(1)}$. This case might not occur for some long videos, e.g., a movie.

For the second example shown in the second row, the query video $q^{(2)}$ shows that a girl is doing her hairstyle in a barbershop. The query video $q^{(2)}$ and its groundtruth video $c_2^{(2)}$ are very similar. The video $c_2^{(2)}$ is a video clip of the original video. The video $c_1^{(2)}$ shows that another girl is doing her hairstyle in a similar barbershop. As the clipping transformation is applied on the videos, the query video and its near-duplicate video are confused with other videos which are not near-duplicate.

For the third example shown in the third row, the query video $q^{(3)}$ shows that two men are shouting. These videos might be a clip of a movie. The query video and its groundtruth video are edited by inserting two different video templates. The content of these two videos is the same. But detecting these near-duplicate videos might be very challenging due to the different video templates.

From these examples, we can see that new challenges and difficulties might be introduced by the new types of near-duplicate videos and hard positive/negative videos in SVD dataset.

Table 3. MAP (%) for hashing based NDVR methods.

Dataset	LSH		ITQ		IsoH		HDML	
	16 bits	32 bits	16 bits	32 bits	16 bits	32 bits	16 bits	32 bits
CCWEB	68.12	83.15	70.16	87.14	72.24	86.75	82.72	90.23
VCDB	10.33	30.88	10.68	33.31	10.60	33.30	35.96	68.92
SVD	4.34	28.36	5.16	30.14	4.85	30.88	6.47	31.59
$\text{SVD}_{Cropping}$	0.32	2.65	0.70	4.41	0.96	4.01	1.23	5.39
$\text{SVD}_{BlackBorder}$	0.76	4.61	1.18	7.08	1.15	5.58	1.61	10.54
$\text{SVD}_{Rotation}$	0.06	0.09	0.04	0.43	0.07	0.24	0.54	1.95
$\text{SVD}_{Speeding}$	3.34	23.56	4.42	25.82	4.14	26.63	4.56	28.60

Table 4. Top-100 MAP (%), storage cost and retrieval time on all datasets.

Methods	Dim/#bits	Top-100 MAP			Storage Cost			Retrieval Time (ms)		
		CCWEB	VCDB	SVD	CCWEB	VCDB	SVD	CCWEB	VCDB	SVD
DML	500D	97.93	84.60	81.27	48.83M	0.40G	2.25G	41.2	278.3	2203.5
CNNL	4096D	97.88	84.48	61.04	99.96M	3.29G	18.42G	266.6	2290.3	15887.3
CNNV	4096D	97.86	79.44	25.10						
LSH+	16 bits	98.29	66.55	76.02	0.06M	0.60M	3.37M	1.4	17.8	88.2
ITQ+		98.11	66.65	77.96						
IsoH+		97.92	66.58	78.19						
HDML+		97.74	77.96	76.29						
LSH+	32 bits	97.81	67.19	78.80	0.09M	0.80M	4.49M	2.5	24.8	174.8
ITQ+		97.75	66.65	78.92						
IsoH+		97.79	67.01	79.00						
HDML+		97.69	78.36	78.63						

5.4. Hashing based NDVR

Accuracy In this section, we present the retrieval results of hashing based methods on all datasets. The MAP results are reported in Table 3. From Table 3, we can find that the retrieval accuracy of hashing based methods are not as good as that of real-value based NDVR methods on all datasets. Compared with CCWEB and VCDB dataset, the retrieval accuracy on SVD dataset is the worst. Furthermore, the MAP results on $\text{SVD}_{transformation}$ are much worse than those on SVD in all cases.

Reranking We also carry out experiments by utilizing reranking to improve the retrieval accuracy of hashing based methods. For reranking, we set $N = 0.1 \times M$, where M is the number of videos in database[7] for each query. Here the videos in database contain labeled videos and background distraction videos or probable negative unlabeled videos

In Table 4, we report the top-100 MAP, storage cost for database and average retrieval time per query. The "LSH+" denotes the LSH algorithm with reranking and the other notations are defined similarly. From Table 4, we can find that after reranking, the retrieval accuracy of hashing based methods is comparable with real-value based methods in most cases. Furthermore, the storage cost for hashing based methods is much smaller than that of real-value based methods. In addition, we can see that hashing based methods are much faster than real-value based methods. Hence, for large-scale applications, hashing based methods are usually more practical than real-value based methods.

[7] As the number of labeled videos for different query videos is different, the M for different query videos is also different.

6. Conclusion

In this paper, we introduce a novel large-scale short video dataset, called SVD, for NDVR. This dataset contains over 500,000 short videos collected from a large video platform and over 30,000 labeled videos of near-duplicate videos. We utilize multiple mining strategies to mine hard positive/negative samples from massive short videos. Furthermore, we design some temporal and spatial transformations to mimic users' copy-and-edit behavior in real applications and construct more challenging variants of SVD. SVD is the first short video dataset, and it is also the largest dataset for NDVR. The release of SVD will foster the research of NDVR, especially NDVR for short videos.

7. Acknowledgement

This work is supported by the NSFC-NRF Joint Research Project (No. 61861146001) and the Program A for Outstanding Ph.D. candidate of Nanjing University. We thank Yubo Du and Ming-Wei Li for their help in data annotation and filtering. Lei Li and Wu-Jun Li are corresponding authors.

References

[1] Yang Cai, Linjun Yang, Wei Ping, Fei Wang, Tao Mei, Xian-Sheng Hua, and Shipeng Li. Million-scale near-duplicate video retrieval system. In *MM*, pages 837–838, 2011.

[2] Chien-Li Chou, Hua-Tsung Chen, and Suh-Yin Lee. Pattern-based near-duplicate video retrieval and localization on web-scale videos. *TMM*, 17(3):382–395, 2015.

[3] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, pages 253–262, 2004.

[4] Matthijs Douze, Herve Jegou, and Cordelia Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *TMM*, 12(4):257–266, 2010.

[5] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824, 2011.

[6] Yanbin Hao, Tingting Mu, Richang Hong, Meng Wang, Ning An, and John Yannis Goulermas. Stochastic multiview hashing for large-scale near-duplicate video retrieval. *TMM*, 19(1):1–14, 2017.

[7] Dong-Chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. *TGRS*, 28(4):509–512, 1990.

[8] Zi Huang, Heng Tao Shen, Jie Shao, Bin Cui, and Xiaofang Zhou. Practical online near-duplicate subsequence detection for continuous video streams. *TMM*, 12(5):386–398, 2010.

[9] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. VCDB: A large-scale database for partial copy detection in videos. In *ECCV*, pages 357–371, 2014.

[10] Yu-Gang Jiang and Chong-Wah Ngo. Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval. *CVIU*, 113(3):405–414, 2009.

[11] Weihao Kong and Wu-Jun Li. Isotropic hashing. In *NeurIPS*, pages 1655–1663, 2012.

[12] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval by aggregating intermediate CNN layers. In *MM*, pages 251–263, 2017.

[13] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *ICCVW*, pages 347–356, 2017.

[14] Julien Law-to, Alexis Joly, and Nozha Boujemaa. Muscle-vcd-2007: a live benchmark for video copy detection, 2007.

[15] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*, pages 1711–1717, 2016.

[16] Jiajun Liu, Zi Huang, HongYun Cai, Heng Tao Shen, Chong-Wah Ngo, and Wei Wang. Near-duplicate video retrieval: Current research and future trends. *CSUR*, 45(4):44:1–44:23, 2013.

[17] Lu Liu, Wei Lai, Xian-Sheng Hua, and Shi-Qiang Yang. Video histogram: A novel video signature for efficient web video duplicate detection. In *MM*, pages 94–103, 2007.

[18] Wu Liu, Tao Mei, and Yongdong Zhang. Instant mobile video search with layered audio-video indexing and progressive transmission. *TMM*, 16(8):2242–2255, 2014.

[19] Ajay Kumar Mallick and Sushila Maheshkar. Near-duplicate video retrieval based on spatiotemporal pattern tree. In *CVIP*, pages 173–186, 2017.

[20] Mohammad Norouzi, David J. Fleet, and Ruslan Salakhutdinov. Hamming distance metric learning. In *NeurIPS*, pages 1070–1078, 2012.

[21] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *PR*, 29(1):51–59, 1996.

[22] Paul Over, George Awad, Jonathan G. Fiscus, Brian Antonishek, Martial Michel, Wessel Kraaij, Alan F. Smeaton, and Georges Quénot. TRECVID 2010 - an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID workshop*, 2010.

[23] Sébastien Poullot, Shunsuke Tsukatani, Phuong Anh Nguyen, Hervé Jégou, and Shin'ichi Satoh. Temporal matching kernel with explicit feature maps. In *MM*, pages 381–390, 2015.

[24] Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, pages 2459–2466, 2013.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[26] Michael W. Schwarz, William B. Cowan, and John C. Beatty. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. *TOG*, 6(2):123–158, 1987.

[27] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, Zhenmin Tang, and Heng Tao Shen. Hashing on nonlinear manifolds. *TIP*, 24(6):1839–1851, 2015.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[29] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *MM*, pages 423–432, 2011.

[30] Ling Wang, Yu Bao, Haojie Li, Xin Fan, and Zhongxuan Luo. Compact CNN based video representation for efficient video copy detection. In *MM*, pages 576–587, 2017.

[31] Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In *NeurIPS*, pages 1753–1760, 2008.

[32] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *MM*, pages 218–227, 2007.

[33] Chuan Xiao, Wei Wang, Xuemin Lin, Jeffrey Xu Yu, and Guoren Wang. Efficient similarity joins for near-duplicate detection. *TODS*, 36(3):15:1–15:41, 2011.

[34] Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. Self-taught hashing for fast similarity search. In *SIGIR*, pages 18–25, 2010.

[35] Wengang Zhou, Houqiang Li, Richang Hong, Yijuan Lu, and Qi Tian. BSIFT: toward data-independent codebook for large scale image search. *TIP*, 24(3):967–979, 2015.