Chapter 4
Finding near-duplicate videos in large-scale collections

Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, Ioannis Kompatsiaris

Abstract This chapter discusses the problem of Near-Duplicate Video Retrieval (NDVR). The main objective of a typical NDVR approach is: given a query video, retrieve all near-duplicate videos in a video repository and rank them based on their similarity to the query. Several approaches have been introduced in the literature, which can be roughly classified in three categories based on the level of video matching, i.e. (i) video-level, (ii) frame-level and (iii) filter-and-refine matching. Two methods based on video-level matching are presented in this chapter. The first is an unsupervised scheme that relies on a modified Bag-of-Word (BoW) video representation. The second is a supervised method based on Deep Metric Learning (DML). For the development of both methods, features are extracted from the intermediate layers of Convolutional Neural Networks and leveraged as frame descriptors, since they offer a compact and informative image representation, and lead to increased system efficiency. Extensive evaluation has been conducted on publicly available benchmark datasets, and the presented methods are compared with state-of-art approaches, achieving the best results in all evaluation setups.

Giorgos Kordopatis-Zilos
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece and School of Electronic Engineering and Computer Science, Queen Mary University, London, UK e-mail: `georgekordopatis@iti.gr`

Symeon Papadopoulos
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece e-mail: `papadop@iti.gr`

Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University, London, UK e-mail: `i.patras@qmul.ac.uk`

Ioannis Kompatsiaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece e-mail: `ikom@iti.gr`

4.1 Introduction

The problem of verifying multimedia content (images, video) that is contributed by users of social media platforms such as YouTube, Instagram and Facebook is of increasing interest given the pervasive use of these platforms and the modern technological capabilities for real-time capturing and sharing of rich multimedia content. For instance, in the context of breaking news events, such as natural disasters or terrorist attacks multiple eyewitness reports are posted and shared through social media platforms. Yet, the reliability and veracity of such reports is often questioned due to the increasing amount of misleading or manipulated content that can quickly spread through online social networks and cause disinformation at large scale. As a result, there is a profound need for technologies that can assist the process of multimedia verification (or the inverse process of debunking fake content).

One popular verification approach adopted by journalists [42] is to try to establish the provenance of a social media post by looking for near-duplicate media items that were posted in the past. For instance, it has been found that images or videos from past events are often re6posted in the context of breaking news events falsely claiming to have been captured in the new setting. For instance, the photo in the tweet in Fig. 4.1 was originally published by the Wall Street Journal on April 2011[1]. However, the same photo was shared thousands of times more than one year later during the Hurricane Sandy (29 Oct 2012), supposedly depicting a dangerous storm descending in New York. To identify such cases and find the origin of an image, journalists often use online services such as Google Images[2] and TinEye[3]. Yet, there is currently no available service or tool to support reverse video search. The research field focusing on this problem is Near-Duplicate Video Retrieval (NDVR).

Due to the exponential growth of social media applications and video sharing websites, NDVR is increasingly important, yet it poses a highly challenging task. At the moment, YouTube reports almost two billions users and more than one billion hours of video viewed per day[4]. Due to the uncontrolled nature of publishing in video platforms, a very common phenomenon is the publication of multiple videos that are either near or partial duplicates of an original video. To this end, our goal is to build an NDVR approach that is able to efficiently and effectively retrieve all such videos in order to support the multimedia verification process.

Being a relatively new research topic, there is a variety of definitions and interpretations of NDVR among the multimedia research community. The definitions vary with respect to the level of resemblance that determines whether a pair of videos are considered related. These range from a very narrow scope, where only the almost identical videos are considered positive pairs [55], to very broad where videos from the same event [38] or with the same semantic content [4] are labeled as related. The definition that is closer to the needs of multimedia verification is the one provided

[1] `https://blogs.wsj.com/metropolis/2011/04/28/weather-journal-clouds-gathered-but-no-tornado-damage/`

[2] `https://images.google.com/`

[3] `https://www.tineye.com/`

[4] `https://www.youtube.com/yt/about/press/` (accessed on March 2019)

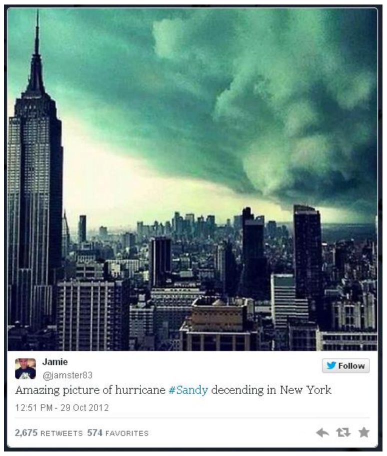

Jamie
@jamster83
Amazing picture of hurricane #Sandy decending in New York
12:51 PM - 29 Oct 2012
2,675 RETWEETS 574 FAVORITES

Fig. 4.1: Tweet example of re-posted image claiming to be breaking news.

in [55]. Based on this, Near-Duplicate videos (NDVs) are considered those that originate from the same source, but can be different in terms of photometric variations (color, light changes), editing operations (caption, logo insertion), encoding parameters (file format), different lengths, or camera settings (camera viewpoint). A number of such NDV examples are presented in Fig. 4.2.

Considerable effort has been invested from the research community to the NDVR problem. Yet, many of the proposed methods are computationally intensive and thus not easy to apply in a setting where many videos need to be verified in very short time. Another limitation is the lack of flexibility, in a sense that near-duplicate search is often too strict (returns only almost exact copies of the input videos) and in some cases it is not catered for the specific requirements of the problem (e.g. when a user needs to look for partial near-duplicate search or for frame-to-video search). Another issue of many state-of-the-art methods is that they adopt a dataset-specific approach: the same dataset is used for both development and evaluation. This leads to specialized solutions that typically exhibit poor performance when used on different video collections.

As a result, the challenge in building an effective NDVR solution is to offer flexibility with respect to the definition of near-duplicate videos and additionally support different requirements of relevance for the verification setting, namely partial-duplicate search), very high precision and recall scores, and at the same time the possibility for scalable indexing massive collections of multimedia and achieving low response times.

Motivated by the excellent performance of deep learning in a wide variety of multimedia problems, we have developed two NDVR approaches that incorporate deep learning and can be used in different application scenarios. For both schemes, we use features from intermediate convolutional layers [40, 60] of pre-trained Convolutional Neural Networks (CNNs) based on the Maximum Activations of Convolu-

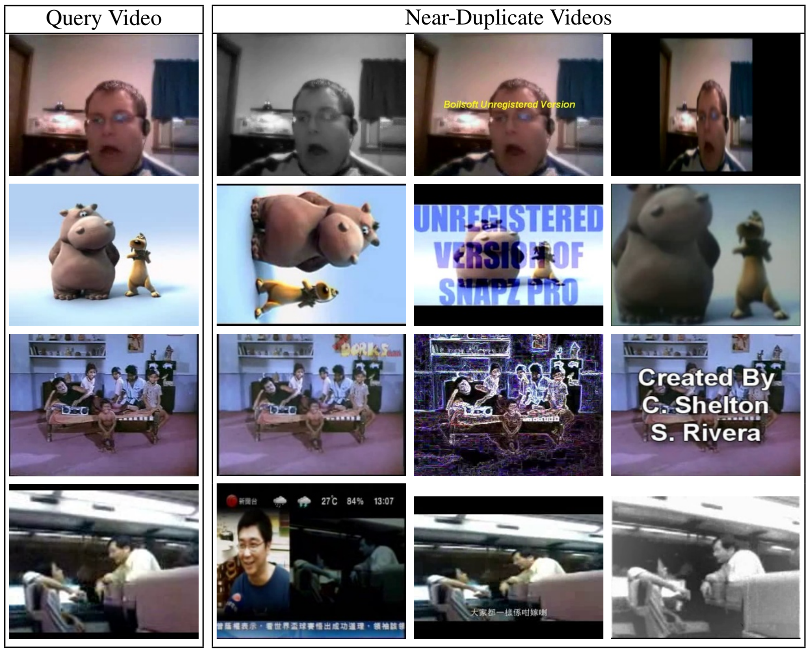

Query Video
Near-Duplicate Videos
Bollsoft Unregistered Version
UNREGISTERED VERSION OF SNAPZ PRO
Created By C. Shelton S. Rivera
27°C 84% 13:07

Fig. 4.2: Examples of queries and near-duplicate videos from the CC_WEB_VIDEO dataset.

tions (MAC). The first approach is unsupervised and is a variation of the traditional Bag-of-Words (BoW) scheme. It uses two layer aggregation techniques, organized in an inverted file structure for fast retrieval. This method does not need labeled data, and as a result it can be applied on any video corpus. However, due to several limitations of this approach (i.e. volatile performance), we also built a second supervised solution leveraging Deep Metric Learning (DML). We set up a DML framework based on a triplet-wise scheme to learn a compact and efficient embedding function. A significant benefit of the learning scheme is that it gives the opportunity to be trained in various scenarios; thus, it provides us with the required flexibility with respect to the NDV definition. Both approaches outperform several state-of-the-art methods on the widely used CC_WEB_VIDEO dataset, and the recently published FIVR-200K dataset.

The reminder of the chapter is organized as follows. In Section 4.2, we review the related literature in the field of NDVR by providing an outline of the major trends in the field. In Section 4.3, we present the two aforementioned NDVR approaches that have been developed within the InVID project [27, 28]. In Section 4.4, we report on the results of a comprehensive experimental study, including a comparison with five state-of-the-art methods. In Section 4.5, we summarize the findings of our work and offer an outlook into future work in the area.

4.2 Literature review

In this section we review several representative works in the literature. NDVR is a challenging problem and has attracted increasing research interest during the last two decades. For a comprehensive overview of the NDVR literature, the reader is referred to Liu et al. [32]. The definition of the NDVR problem is discussed and compared to those of similar research problems (Section 4.2.1). Additionally, a variety of approaches are presented and classified based on the type of similarity computed between videos (Section 4.2.2). Finally, several publicly available benchmark datasets used to evaluate such approaches are described (Section 4.2.3).

4.2.1 Definition and related research problems

There is a variety of definitions and interpretations among the multimedia research community regarding the concept of NDVs, as pointed in [33]. The representative and predominant definitions are those proposed in Wu et al. [55], Shen et al. [41] and Basharat et al. [4]. These vary with respect to the level of resemblance that determines whether a pair of videos are considered to be near-duplicates.

Wu et al. [55] adopted the most narrow scope among the definitions: *NDVs were considered only those that are identical or approximately identical videos, i.e. close to being exact duplicates of each other, but different in terms of file format, encoding parameters, minor photometric variations, editing operations, length, and other modifications*. By contrast the definition in Shen et al. [41] extended this to videos with *the same semantic content but different in various aspects introduced during capturing time, including photometric or geometric settings*. Another definition was suggested by Basharat et al. [4], which *considered NDVs as videos originating from the same scene*. The same semantic concept can occur under different illumination, appearance, scene settings, camera motion, etc.

Cherubini et al. [8] conducted a large-scale online survey to formulate the definition of NDVs based on the human perception. The results revealed that the technical definitions with respect to manipulations of visual content in Wu et al. [55] and Shen et al. [41] agree to the human perception. However, videos differing with respect to overlaid or added visual content were not perceived as near-duplicates. It is evidenced that users perceive as near-duplicate those videos that are both visually similar and semantically very close [8].

Additionally, NDVR is closely related with other research fields, such as Video Copy Detection (VCD) [31] and Event Video Retrieval (EVR) [38]. The definition of video copies in VCD is very close to the one of NDVR, yet it is slightly narrower. Videos derived from the same source video and differing only with respect to photometric or geometric transformations are considered as copies based on Law-To et al. [31]. Also, the objective of a VCD approach is to identify the copied videos and detect the particular video segments that have been copied. The EVR problem was formulated by Revaud et al. [38]. The objective of EVR is the retrieval of videos that

Research	Video rep.	mAP
Wu et al., 2009 [55]	GV	0.892
Shang et al., 2010 [39]	GV	0.953
Song et al., 2013 [46]	HC	0.958
Hao et al., 2017 [12]	HC	0.971
Jing et al., 2018 [23]	HC	0.972

Table 4.1: Video representation and mean Average Precision (mAP) on CC_WEB_VIDEO dataset of five video-level matching methods. GV stands for global vectors, HC for hash codes.

capture the same event. The definition of same-event videos is very broad, including videos that have either spatial or temporal relationship.

4.2.2 NDVR approaches

The NDVR approaches can be classified based on the level of matching performed to determine near-duplicity into video-level (Section 4.2.2.1), frame-level (Section 4.2.2.2) and filter-and-refine matching (Section 4.2.2.3)

4.2.2.1 Video-level matching

Video-level approaches have been developed to deal with web-scale retrieval. In such approaches, videos are usually represented with a global signature such as an aggregated feature vector [55, 34, 16, 39, 6] or a hash code [45, 46, 12, 11, 23]. The video matching is based on the similarity computation between the video representations. Table 4.1 displays the performance of five video-level approaches on CC_WEB_VIDEO.

A common process to generate a video representation is by the combination of visual features extracted from the video frames into a single feature vector. Wu et al. [55] introduced a simple approach for the video signature generation. They extracted HSV features from the video keyframes and averaged them to create a single vector. The distance between two video signatures was computed based on their Euclidean distance. Huang et al. [16] proposed a video representation model called Bounded Coordinate System (BCS), which extended Principal Component Analysis (PCA) over the colour histograms of the video frames. To compute the similarity between two BCS signatures, scaling and rotation were both considered for matching videos. To improve retrieval efficiency, a two-dimensional transformation method was introduced based on the bi-directional axes in BCS. Liu et al. [34] proposed a method where each video was compared with a set of seed vectors, derived from a number of reference videos. The percentage of video frames that were close to the corresponding reference video was calculated based on the Euclidean distance of their colour histograms and used to determine the video similarity. Shang et al. [39] introduced compact spatio-temporal features based on Local Binary Patterns (LBP) [59], called STF-LBP, to represent videos and constructed a modified inverted file index. These spatio-temporal features were extracted based on a feature selection and *w*-shingling scheme. They adopted Jaccard similarity to rank videos. Cai et al. [6] presented a large-scale BoW approach by applying a scalable K-means clustering technique on the color correlograms [14] of a sample of frames and using inverted file indexing [44] for the fast retrieval of candidate videos. They used cosine similarity to measure similarity between two candidate videos.

Hashing schemes have been extensively used for NDVR. Song et al. [45] presented an approach for Multiple Feature Hashing (MFH) based on a supervised method that employed multiple frame features (i.e. LBP and HSV features) and learned a group of hash functions that map the video keyframe descriptors into the Hamming space. The video signatures were generated by averaging the keyframe hash codes. The Hamming distance was employed to calculate video distances. They extended their approach in [46] by including information of the keyframe groups into the objective function, so as to introduce temporal information in the learning process of the hash functions, which led to a marginal performance increase. Hao et al. [12] combined multiple keyframe features to learn a group of mapping functions that projected the video keyframes into the Hamming space. The combination of the keyframe hash codes generated a video signature that constituted the video representation in the dataset. The Kullback-Leibler (KL) divergence measure was used to approximate the retrieval scores. They extended their work in [11] by employing t-distribution to estimate the similarity between the relaxed hash codes and introduced a deep hashing architecture based on a 3-layer CNN. Jing et al. [23] proposed a supervised hashing method called Global-View Hashing (GVH), which utilized relations among multiple features of video keyframes. They projected all features into a common space and learned multi-bit hash codes for each video using only one hash function. The Hamming distance of the learned hash codes was used to rank the retrieved videos with respect to an input query.

4.2.2.2 Frame-level matching

In the case of frame-level matching approaches, the near-duplicate videos are determined by the comparison between individual video frames or sequences. Typical frame-level approaches [48, 10, 32, 22, 54] calculate the frame-by-frame similarity and then employ sequence alignment algorithms to compute similarity at the video level. Moreover, a lot of research effort has been invested in methods that exploit spatio-temporal features to represent video segments in order to facilitate video-level similarity computation [15, 56, 38, 37, 3]. Table 4.2 displays the performance of four approaches on the VCDB dataset.

Frame-level methods usually extract local information of video frames and generate a global frame representation for similarity calculation and video matching. Tan et al. [48] introduced an approach based on Temporal Networks (TN). They

Research	**Features**	**F1 score**
Jiang et al., 2014 [21]	HC	60.0%
Jiang et al., 2016 [22]	DL	65.0%
Wang et al., 2017 [54]	DL	70.4%
Baraldi et al., 2018 [3]	DL+ST	68.7%

Table 4.2: Employed features and F1 score (%) on VCDB of four frame-level matching methods. HC stands for hand-crafted, DL for deep learning and ST for spatio-temporal features.

embedded temporal constrains into a network structure and formulated the partial video alignment problem into a network flow problem. The near-duplicate video segments were determined based on the longest network's paths. Also, to precisely decide the boundaries of the overlapping segments, pair-wise constraints generated from keypoint matching were applied. Douze et al. [10] detected points of interest using the Hessian-Affine region detector [36], and extracted SIFT [35] and CS-LBP [13] descriptors, in order to create a BoW codebook [44] for Hamming Embedding with weak geometric consistency [17]. Using post-filtering, they verified retrieved matches with spatio-temporal constrains and devised the so-called temporal Hough Voting (HV). Jiang et al. [22] employed a pre-trained CNN to extract global features for the video frames and they also trained another CNN with pairs of image patches that captures the local information of frames. They experimented with TN and HV in order to detect the copied video segments. Wang et al. [54] proposed a compact video representation by combining features extracted from pre-trained CNN architectures with sparse coding to encode them into a fixed length vector. To determine the copied video segments, they constructed TNs based on the distance between the extracted features.

Some works utilized spatio-temporal features to accelerate the matching process and improve the performance by considering not only the spatial information of frames but also the temporal relations among frames. Huang et al. [15] proposed a one-dimensional Video Distance Trajectory (VDT) based on the continuous changes of consecutive frames with respect to a reference point. VDT was further segmented and represented by a sequence of compact signatures called Linear Smoothing Functions (LSFs), which utilized the compound probability to combine three independent video factors and compute sequence similarity. Wu and Aizawa [56] proposed a self-similarity-based feature representation called Self-Similarity Belt (SSBelt), which derived from the Self-Similarity Matrix (SSM). The interest corners were detected and described by a BoW representation. Revaud et al. [38] proposed the Circulant Temporal Encoding (CTE) that encodes frame features in a spatio-temporal representation with Fourier transform. The videos are compared in the frequency domain based on the properties of circulant matrices. Poullot et al. [37] introduced the Temporal Matching Kernel (TMK) that encodes sequences of frames with periodic kernels that take into account the frame descriptor and timestamp. A score function was introduced for video matching that maximizes both the similarity score and the

Research	Multimodal	F1 score
Tian et al., 2013 [50]	✓	0.950
Jiang et al., 2012 [20]	✓	0.962
Tian et al., 2015 [52]	✓	0.952
Chou et al., 2015 [9]	✗	0.938

Table 4.3: Multimodal approach and F1 score on TRECVID 2011 of four filter-and-refine matching methods. If the approach is not multimodal, then the F1 score is calculated based on the video transformations only.

relative time offset by considering all possible relative timestamps. Baraldi et al. [3] built a deep learning layer component based on TMK and set up a training process to learn the feature transform coefficients in the Fourier domain. A triplet loss that takes into account both the video similarity score and the temporal alignment was used in order to train the proposed network.

4.2.2.3 Filter-and-refine matching

To overcome the bottleneck of video-level approaches and to achieve efficient NDVR implementations, researchers developed hybrid approaches by combining the advantages of frame-level and video-level methods. Table 4.3 displays the performance of four filter-and-refine approaches on TRECVID 2011. Wu et al. [55] generated video signatures by averaging the HSV histograms of keyframes. Then, they applied a hierarchical filter-and-refine scheme to cluster and filter out near-duplicate videos. When a video could not be clearly classified as NDV, they calculated video similarity based on an expensive local feature-based scheme. Tian et al. [51] extracted audio-visual features. They applied a BoW scheme on the local visual features (SIFT [35], SURF [5]) and a locality sensitive hashing (LSH) scheme on global visual features (DCT) and audio features (WASF [7]). A sequential pyramid matching (SPM) algorithm was devised to localize the similar video sequences. In contrast, Jiang et al. [20] presented a soft cascade framework utilizing multiple hashed features to filter out non-NDVs. They modified the SPM to introduce temporal information in a temporal pyramid matching (TPM). To further improve performance, they proposed in [50] a multi-scale sequence matching method by LSH using WASF, DCT, and the dense color version SIFT (DC-SIFT), combined with TPM to match near-duplicate segments. Including the concept of transformation-awareness, copy units, and soft decision boundary, Tian et al. [52] extended the multimodal detector cascading framework [20], [50] to a more general approach. Chou et al. [9] proposed a spatio-temporal indexing structure utilizing index patterns, termed Pattern-based Index Tree (PI-tree), to early filter non-near-duplicate videos. In the refine stage, an m-Pattern-based Dynamic Programming scheme was devised to localize near-duplicate segments and to re-rank results of the filter stage. Yang et al. [57] proposed a multi-scale video sequence matching method, which

Dataset	**Queries**	**Videos**	**User-gen.**	**Retrieval Task**
CC_WEB_VIDEO [55]	24	12,790	✓	Near-Duplicate Video Retrieval
UQ_VIDEO [45]	24	169,952	✓	Near-Duplicate Video Retrieval
MUSCLE-VCD [31]	18	101	✗	Video Copy Detection
TRECVID 2011 [29]	11,256	11,503	✗	Video Copy Detection
VCDB [19]	528	100,528	✓	Partial Video Copy Detection
EVVE [38]	620	102,375	✓	Event Video Retrieval
FIVR-200K [26]	100	225,960	✓	Fine-grained Incident Video Retrieval

Table 4.4: Publicly available video datasets developed for retrieval tasks related to NDVR.

gradually detected and located similar segments between videos from coarse to fine scales. Given a query, they used a maximum weight matching algorithm to rapidly select candidate videos in the coarser scale, then extracted the similar segments in the middle scale to find the NDVs. In the fine scale, they used bi-directional scanning to check the matching similarity of video parts to localize near-duplicate segments.

4.2.3 Benchmark datasets

Although the problem of NDVR has been investigated for at least two decades, few benchmark datasets have been published. Table 4.4 presents an overview of several publicly available datasets developed for related retrieval tasks. Many researchers constructed their own datasets and did not release them. For instance, Shen et al. [41] collected and manually annotated more than 11,000 TV commercials with an average length of about 60 seconds.

The most popular and publicly available dataset related to the NDVR problem is CC_WEB_VIDEO [55]. It was published by the research groups of City University of Hong Kong and Carnegie Mellon University and consists of 13,129 generated videos collected from the Internet. For the dataset collection, 24 popular text queries were submitted to popular video platforms, such as YouTube, Google Video, and Yahoo! Video. A set of videos were collected for each query and the video with the most views was selected as the query video. Then, videos in the collected sets were manually annotated based on their relation to the query video. It is noteworthy that video sets contain high amounts of near-duplicates. On average there are 27% videos per query that are considered near-duplicates to the most popular version of a video in the search results. However, for certain queries, the redundancy of non near-duplicates can be as high as 94%.

Several variations of the CC_WEB_VIDEO dataset were developed [39, 45, 6, 9]. In order to make the NDVR problem more challenging and benchmark the scalability of their approaches, researchers usually extend the core CC_WEB_VIDEO dataset with many thousands of distractor videos. The most well-known public dataset that was created through this process is UQ_VIDEO [45]. For the dataset collection, they chose the 400 most popular queries based on Google Zeitgeist Archives from years 2004 to 2009. Each query was fed to YouTube search and they limited the returned videos to one thousand. After filtering out videos with size greater than 10MB, the combined dataset contains 169,952 videos (including those of the CC_WEB_VIDEO) in total with 3,305,525 keyframes and the same 24 query videos contained in CC_WEB_VIDEO. Unfortunately, only the HSV and LBP histograms of the video keyframes are provided by the authors.

Another popular public benchmark is the Muscle-VCD, created by Law-To et al. [31]. This dataset was designed for the problem of VCD. It consists of 100 hours of videos that include Web video clips, TV archives, and movies with different bitrates, resolutions and video format. A set of original videos and their corresponding transformed videos are given for the evaluation of copy detection algorithms. Two kinds of transformation were applied on the queries: a) entire video copy with a single transformation, where the videos may be slightly recoded and/or noised; b) partial video copy with a mixture of transformations, where two videos only share one or more short segments. Both transformations were simulated by using video-editing software to apply the transformations. The transformed videos or segments were used as queries to search their original versions in the dataset.

The annual TRECVID [1] evaluation included a task on copy detection in years 2008 to 2011. Each year a benchmark dataset was generated and released only to the registered participants of the task. The TRECVID datasets were constructed in a very similar way to the Muscle-VCD dataset. The latest edition of the dataset [29] contained 11,503 reference videos of over 420 hours. Query videos were categorized into three types: (i) a reference video only, (ii) a reference video embedded into a non-reference video, and (iii) a non-reference video only. Only the first two types of query videos were near-duplicates to videos in the dataset. Each query was generated using a software to randomly extract a segment from a dataset video and impose a few predefined transformations. The contestants were ask to find the original videos and detect the copied segment.

A more recent dataset that is relevant to our problem is the VCDB [19]. This dataset is composed of videos derived from popular video platforms (i.e. YouTube and Metacafe) and has been compiled and annotated as a benchmark for the partial copy detection problem, which is highly related to the NDVR problem. VCDB contains two subsets, the core and the distractor subset. The core subset contains 28 discrete sets of videos composed of 528 query videos and over 9,000 pairs of partial copies. Each video set was manually annotated by seven annotators and the video chunks of the video copies were extracted. The distractor subset is a corpus of approximately 100,000 distractor videos that is used to make the video copy detection problem more challenging.

Moreover, the EVVE (EVent VidEo) [38] dataset was developed for the problem of event video retrieval. The main objective of the systems evaluated on this dataset is the retrieval of all videos that capture a particular event given a query video. The dataset contains 13 major events that were provided as queries to YouTube. A total number of 2,375 videos were collected and 620 of them were selected as video queries. Each event was annotated by one annotator, who first produced a precise definition of the event. In addition to the videos collected for specific events, the authors also retrieved a set of 100,000 distractor videos by querying YouTube with unrelated terms. These videos were collected before a certain date to ensure that the distractor set did not contain any of the relevant events of EVVE, since all events were temporally localized after that date.

Finally, the FIVR-200K [26] dataset was developed to simulate the problem of Fine-grained Incident Video Retrieval (FIVR). For the dataset collection, the major events occurring in the time span from January 2013 to December 2017 were collected by crawling Wikipedia. The event headlines were then used to query YouTube. In total, 225,960 videos were collected from 4,687 events, and 100 query videos were selected using a systematic process. Then the videos in the dataset were manually annotated based on their relation to the queries. FIVR-200K includes three different tasks: a) the Duplicate Scene Video Retrieval (DSVR) task which is highly related to the NDVR problem and it only accepts as positive matches videos that contain at least one identical scene, b) the Complementary Scene Video Retrieval (CSVR) task which is a broader variant of the NDVR problem where videos that contain scenes captured at the same time but from different camera viewpoints are considered related, and c) Incident Scene Video Retrieval (ISVR) task where all videos with scenes displaying the same incident are considered positive matches.

4.3 NDVR approaches in InVID

In InVID, two video-level solutions have been developed, since video-level NDVR appeared to offer the best trade-off between computational cost and retrieval effectiveness. Additionally, most video-level methods can be adapted to a corresponding frame-level approach in a straightforward manner if further retrieval accuracy is needed. Of the two developed video-level approaches, one is unsupervised (Section 4.3.1) and the other supervised (Section 4.3.2). The former is a modified BoW-scheme based on the extracted CNN features. The latter is based on Deep Metric Learning (DML), which learns an embedding function that maps the CNN descriptors into a feature space where the NDVs are closer than the other videos.

4.3.1 Bag-of-Words approach

The proposed unsupervised NDVR approach relies on a Bag-of-Words (BoW) scheme [27]. In particular, two aggregation variations are proposed: a vector aggregation where a single codebook of visual words is used, and a layer aggregation where multiple codebooks of visual words are used. The video representations are organised in an inverted file structure for fast indexing and retrieval. Video similarity is computed based on the cosine similarity of the *tf-idf* weighted vectors of the extracted BoW representations.

4.3.1.1 CNN-based feature extraction

In recent research [58, 60], pre-trained CNN models are used to extract visual features from intermediate convolutional layers. These features are computed through the forward propagation of an image over the CNN network and the use of an aggregation function (e.g. VLAD encoding [18], max/average pooling) on the convolutional layer.

We adopt a compact representation for frame descriptors that is derived from activations of all intermediate convolutional layers of a pre-trained CNN by applying the function called Maximum Activation of Convolutions (MAC) [40, 60]. A pre-trained CNN network $\mathscr{C}$ is considered, with a total number of L convolutional layers, denoted as $\mathscr{L}^1, \mathscr{L}^2, ..., \mathscr{L}^L$. Forward propagating a frame through network $\mathscr{C}$ generates a total of L feature maps, denoted as $\mathscr{M}^l \in \mathbb{R}^{n_d^l \times n_d^l \times c^l}$ $(l = 1, ..., L)$, where $n_d^l \times n_d^l$ is the dimension of every channel for convolutional layer $\mathscr{L}^l$ (which depends on the size of the input frame) and c^l is the total number of channels. To extract a single descriptor vector from every layer, an aggregation function is applied on the above feature maps. In particular, we apply max pooling on every channel of feature map $\mathscr{M}^l$ to extract a single value. The extraction process is formulated in:

$$v^l(i) = \max\ \mathscr{M}^l(\cdot, \cdot, i), \quad i = \{1, 2, ..., c^l\} \tag{4.1}$$

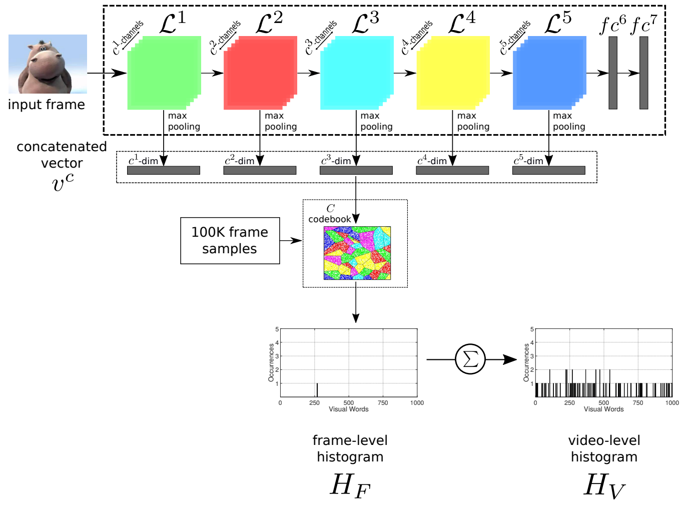
where layer vector v^l is a c^l-dimensional vector that is derived from max pooling on every channel of feature map $\mathscr{M}^l$. The layer vectors are then ℓ_2-normalized.

We extract and concatenate frame descriptors only from activations in intermediate layers, since we aim to construct a visual representation that preserves local structure at different scales. Activations from fully-connected layers are not used, since they are considered to offer a global representation of the input. A positive side-effect of this decision is that the resulting descriptor is compact, reducing the total processing time and storage requirements. For the VGGNet and GoogLeNet architectures, we do not use the initial layer activations as features, since those layers are expected to capture very primitive frame features (e.g. edges, corners, etc.) that could lead to false matches.

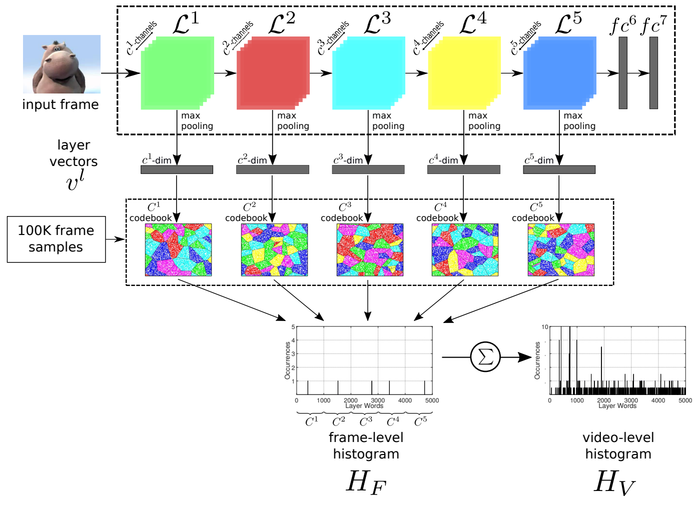
Uniform sampling is applied to select one frame per second for every video and extract the respective features for each of them. Hence, given an arbitrary video with N frames $\{F_1, F_2, ..., F_N\}$, the video representation is a set that contains all feature vectors of the video frames $v = \{v_{F_1}, v_{F_2}, ..., v_{F_N}\}$, where v_{F_i} contains all layer vectors of frame F_i. Although v_{F_i} stands for a set of vectors, we opted to use this notation for convenience.

4.3.1.2 Feature aggregation

We follow two alternative feature aggregation schemes (i.e. ways of aggregating features from layers into a single descriptor for the whole frame): a) *vector aggregation* and b) *layer aggregation*. The outcome of both schemes is a frame-level histogram



(a) Vector Aggregation



(b) Layer Aggregation

Fig. 4.3: The two proposed aggregation schemes and the final video representation.

H_F that is considered as the representation of a frame. Next, a video-level histogram H_V is derived from the frame representations by aggregating frame-level histograms to a single video representation. Figure 4.3 illustrates the two schemes.

Vector aggregation: A bag-of-words scheme is applied on the vector v^c resulting from the concatenation of individual layer features to generate a single codebook of K visual words, denoted as $C_K = \{t_1, t_2, ..., t_K\}$. The selection of K, a system parameter, has critical impact on the performance of the approach. Having generated the visual codebook, every video frame is assigned to the nearest visual word. Accordingly, every frame F_i with feature descriptor $v^c_{F_i}$ is aggregated to the nearest visual word $t_{F_i} = NN(v^c_{F_i})$, hence its H_{F_i} contains only a single visual word.

Layer aggregation: To preserve the structural information captured by intermediate layers $\mathscr{L}$ of the CNN network $\mathscr{C}$, we generate L layer-specific codebooks of K words (denoted as $C^l_K = \{t^l_1, t^l_2, ..., t^l_K\}, l = 1, ..., L$), which we then use to extract separate bag-of-words representations (one per layer). The layer vectors $v^l_{F_i}$ of frame F_i are mapped to the nearest layer words $t^l_{F_i} = NN(v^l_{F_i})$, $(l = 1, 2, ..., L)$. In contrast to the previous scheme, every frame F_i is represented by a frame-level histogram H_{F_i} that results from the concatenation of the individual layer-specific histograms, thus comprising L words instead of a single one.

The final video representation is generated based on the BoW representations of its frames. In particular, given an arbitrary video with N frames $\{F_1, F_2, ..., F_N\}$, its video-level histogram H_V is derived by summing the histogram vectors corresponding to its frames, i.e. $H_V = \sum_{i \in [1,N]} H_{F_i}$. Note that for the two aggregation schemes, histograms of different sizes are generated. In the first case, the total number of visual words equals K, whereas in the second case, it equals $K \cdot L$.

4.3.1.3 Video indexing and querying

In the proposed approach, we use *tf-idf* weighting to calculate the similarity between two video histograms. The *tf-idf* weights are computed for every visual word in every video in a video collection C_b:

$$w_{td} = n_{td} \cdot \log |C_b| / n_t \tag{4.2}$$

where w_{td} is the weight of word t in video d, n_{td} and n_t are the number of occurrences of word t in video d and the entire collection respectively, while $|C_b|$ is the number of videos in the collection. The former factor of the equation is called *term frequency* (tf) and the latter is called *inverted document frequency* (idf).

Video querying is the online part of the approach. Let q denote a query video. Once the final histogram H^q_v is extracted from the query video, an inverted file indexing scheme [44] is employed for fast and efficient retrieval of videos that have at least a common visual word with the query video. For all these videos (i.e. videos with non-zero similarity), the cosine similarity between the respective *tf-idf* representations is computed:

$$sim(q,p) = \frac{\mathbf{w}_q \cdot \mathbf{w}_p}{\|\mathbf{w}_q\| \, \|\mathbf{w}_p\|} = \frac{\sum_{i=0}^{K} w_{iq} w_{ip}}{\sqrt{\sum_{i=0}^{K} w^2_{iq}} \sqrt{\sum_{i=0}^{K} w^2_{ip}}} \tag{4.3}$$

where $\mathbf{w}_q$ and $\mathbf{w}_p$ are the weight vectors of videos q and p, respectively, and $\|\mathbf{w}\|$ is the norm of vector $\mathbf{w}$. The collection videos are ranked in descending order based on their similarity to the query.

In the inverted file structure, each entry corresponds to a visual word, and contains its ID, the *idf* value and all the video IDs in which the visual word occurs. The video IDs map to a video in the collection C_b where the occurrences (*tf*) of the visual words are stored. With this inverted file structure, all the needed values for the calculation of the similarity between a query and a dataset video can be retrieved.

4.3.2 Deep Metric Learning approach

The unsupervised approach has several limitations. The most important is that it offers a dataset-specific solution, i.e. the extracted knowledge is not transferable, and re-building the model is computationally expensive. To observe no performance loss, a sufficiently large and diverse dataset to create vocabularies is required, which needs significant effort to be collected or sometimes is not even possible. Hence, we also developed a Deep Metric Learning (DML) approach to overcome this limitation [28]. This involves training a Deep Neural Network (DNN) to approximate an embedding function for the accurate computation of similarity between two candidate videos. For training, we devised a novel triplet generation process.

For feature extraction, we build upon the same process as the one presented in Section 4.3.1.2. Hence, given an arbitrary video with N frames $\{F_1, F_2, ..., F_N\}$, we extract one feature descriptor for each video frame by concatenating the layer vector to a single vector. Global video representations v are then derived by averaging and normalizing (zero-mean and ℓ_2-normalization) these frame descriptors. Keep in mind that feature extraction is not part of the training (deep metric learning) process, i.e. the training of the network is not end-to-end, and as a result the weights of the pre-trained network used for feature extraction are not updated.

4.3.2.1 Problem formulation

We address the problem of learning a pairwise similarity function for NDVR from the relative information of pairwise/triplet-wise video relations. For a given query video and a set of candidate videos, the goal is to quantify the similarity between the query and every candidate video and use it for the ranking of the entire set of candidates with the goal of retrieving the NDVs at the top ranks. We first define the similarity between two arbitrary videos q and p as the squared Euclidean distance in the video embedding space:

$$\mathrm{D}(f_\theta(q), f_\theta(p)) = \|f_\theta(q) - f_\theta(p)\|_2^2 \tag{4.4}$$

where, $f_\theta(\cdot)$ is the embedding function that maps a video to a point in the Euclidean space, θ are the system parameters and $\mathrm{D}(\cdot,\cdot)$ is the squared Euclidean distance in this space. Additionally, we define a pairwise indicator function $\mathrm{I}(\cdot,\cdot)$ that specifies whether a pair of videos are near-duplicate.

$$\mathrm{I}(q,p) = \begin{cases} 1 & \text{if } q,p \text{ are NDVs} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Our objective is to learn an embedding function $f_\theta(\cdot)$ that assigns smaller distances to NDV pairs than others. Given a video with feature vector v, a NDV with v^+ and a dissimilar video with v^-, the embedding function $f_\theta(\cdot)$ should map video vectors to a common space $\mathbb{R}^d$, where d is the dimension of the feature embedding, in which the distance between query and positive videos is always smaller than the distance between query and negative. This is formulated as:

$$\begin{aligned} &\mathrm{D}(f_\theta(v), f_\theta(v^+)) < \mathrm{D}(f_\theta(v), f_\theta(v^-)), \\ &\forall v, v^+, v^- \text{ such that } \mathrm{I}(v,v^+) = 1, \mathrm{I}(v,v^-) = 0 \end{aligned} \tag{4.6}$$

4.3.2.2 Triplet loss

To implement the learning process, we create a collection of N training instances organized in the form of triplets $\mathscr{T} = \{(v_i, v_i^+, v_i^-), i = 1,...,N\}$, where v_i, v_i^+, v_i^- are the feature vectors of the query, positive (NDV), and negative (dissimilar) videos. A triplet expresses a relative similarity order among three videos, i.e. v_i is more similar to v_i^+ in contrast to v_i^-. We define the following hinge loss function for a given triplet called ‘triplet loss’ :

$$L_\theta(v_i, v_i^+, v_i^-) = \max\{0, \mathrm{D}(f_\theta(v_i), f_\theta(v_i^+)) - \mathrm{D}(f_\theta(v_i), f_\theta(v_i^-)) + \gamma\} \tag{4.7}$$

where γ is a margin parameter to ensure a sufficiently large difference between the positive-query distance and negative-query distance. If the video distances are calculated correctly within margin γ, then this triplet will not be penalised. Otherwise the loss is a convex approximation of the loss that measures the degree of violation of the desired distance between the video pairs specified by the triplet. To this end, we use batch gradient descent to optimize the objective function:

$$\min_\theta \sum_{i=1}^{m} L_\theta(v_i, v_i^+, v_i^-) + \lambda \left\|\theta\right\|_2^2 \tag{4.8}$$

where λ is a regularization parameter to prevent overfitting, and m is the total size of a triplet mini-batch. Minimising this loss will narrow the query-positive distance while widening the query-negative distance, and thus lead to a representation satisfying the desirable ranking order. With an appropriate triplet generation strategy

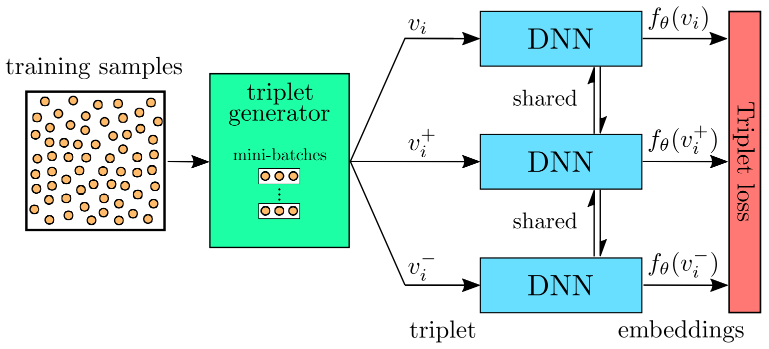


Fig. 4.4: Illustration of DML network architecture.

in place, the model will eventually learn a video representation that improves the effectiveness of the NDVR solution.

4.3.2.3 DML network architecture

For training the DML model, a triplet-based network architecture is proposed (Fig. 4.4) that optimizes the triplet loss function in Equation 4.7. The network is provided with a set of triplets $\mathscr{T}$ created by the triplet generation process of Section 4.3.2.5. Each triplet contains a query, a positive and a negative video with v_i, v_i^+ and v_i^- feature vectors, respectively, which are fed independently into three siamese DNNs with identical architecture and parameters. The DNNs compute the embeddings of $v : f_\theta(v) \in \mathbb{R}^d$. The architecture of the deployed DNNs is based on three dense *fully-connected layers* and a *normalization layer* at the end leading to vectors that lie on a d-dimensional unit length hypersphere, i.e. $\|f_\theta(v)\|_2 = 1$. The size of each hidden layer (number of neurons) and the d-dimension of the output vector $f_\theta(v)$ depends on the dimensionality of input vectors, which is in turn dictated by the employed CNN architecture. The video embeddings computed from a batch of triplets are then given to a triplet loss layer to calculate the accumulated cost based on Equation 4.7.

4.3.2.4 Video-level similarity computation

The learned embedding function $f_\theta(\cdot)$ is used for computing similarities between videos in a target video corpus. Given an arbitrary video with $v = \{v_{F_1}, v_{F_2}, ..., v_{F_N}\}$, two variants are proposed for fusing similarity computation across video frames: early and late fusion.

Early fusion: Frame descriptors are averaged and normalized into a global video descriptor before they are forward propagated to the network. The global video signature is the output of the embedding function $f_\theta(\cdot)$:

$$f_\theta(v) = f_\theta(\frac{1}{N}\sum_{i=1}^{N} v_{F_i}) \tag{4.9}$$

Late fusion: Each extracted frame descriptor of the input video is fed forward to the network, and the set of their embedding transformations is averaged and normalized:

$$f_\theta(v) = \frac{1}{N}\sum_{i=1}^{N} f_\theta(v_{F_i}) \tag{4.10}$$

There are several pros and cons for each scheme. The former is computationally lighter and more intuitive; however, it is slightly less effective. Late fusion leads to better performance and is amenable to possible extensions of the base approach (i.e. frame-level approaches). Nonetheless, it is slower since the features extracted from all selected video frames are fed to the DNN.

Finally, the similarity between two videos derives from the distance of their representations. For a given query q and a set of M candidate videos $\{p_i\}_{i=1}^{M} \in P$, the similarity within each candidate pair is determined by normalizing the distance with respect to the maximum value and then subtracting the result from the unit to map the similarity scores to the range [0, 1]. This process is formulated in:

$$\mathrm{S}(q,p) = 1 - \frac{\mathrm{D}(f_\theta(q), f_\theta(p))}{\max\limits_{p_i \in P}(\mathrm{D}(f_\theta(q), f_\theta(p_i)))} \tag{4.11}$$

where $S(\cdot,\cdot)$ the similarity between two videos and $\max(\cdot)$ is the maximum function.

4.3.2.5 Triplet generation

A crucial part of the proposed approach is the generation of the video triplets. It is important to provide a considerable amount of videos for constructing a representative triplet training set. However, the total number of triplets that can be generated equals the total number of 3-combinations over the size N of the video corpus, i.e.:

$$\binom{N}{3} = \frac{N\cdot(N-1)\cdot(N-2)}{6} \tag{4.12}$$

We have empirically determined that only a tiny portion of videos in a corpus could be considered as near-duplicates for a given video query. Thus, it would be inefficient to randomly select video triplets from this vast set (for instance, for $N = 1,000$, the total number of triplets would exceed 160M). Instead, a sampling strategy is employed as a key element of the triplet generation process, which is focused on selecting hard candidates to create triplets.

The proposed sampling strategy is applied on a development dataset. Such a dataset needs to contain two sets of videos: $\mathscr{P}$, a set of near duplicate video pairs that are used as query-positive pairs, and $\mathscr{N}$, a set of dissimilar videos that are used as negatives. We aim at generating *hard triplets*, i.e. negative videos (*hard negatives*) with distance to the query that is smaller than the distance between the query and positive videos (*hard positives*). The aforementioned condition is expressed in:

$$\mathscr{T} = \{(q,p,n) | (q,p) \in \mathscr{P}, n \in \mathscr{N}, \mathrm{D}(q,p) > \mathrm{D}(q,n)\} \tag{4.13}$$

where $\mathscr{T}$ is the resulting set of triplets. The global video features are first extracted following the process in Section 4.3.1.1. Then, the distance between every query in $\mathscr{P}$ and every dissimilar video in $\mathscr{N}$ is calculated. If the query-positive distance is greater than a query-negative distance, then a hard triplet is formed composed of the three videos. The distance is calculated based on the Euclidean distance of the initial global video descriptors.

4.4 Evaluation

In this section, the two developed approaches are evaluated. The experimental setup is described in Section 4.4.1, where we present the datasets used, the evaluation metrics, several implementation details, and a number of competing approaches from the state-of-the-art. Extensive experimental evaluation is conducted and reported under various evaluation settings in Section 4.4.2.

4.4.1 Experimental setup

4.4.1.1 Evaluation datasets

Experiments were performed on the CC_WEB_VIDEO dataset [55], which is available by the research groups of City University of Hong Kong and Carnegie Mellon University. The collection consists of a sample of videos retrieved by submitting 24 popular text queries to popular video sharing websites (i.e. YouTube, Google Video, and Yahoo! Video). For every query, a set of video clips were collected and the most popular video was considered to be the query video. Subsequently, all videos in the collected set were manually annotated based on their near-duplicate relation to the query video. Table 4.5 depicts the types of near-duplicate types and their annotation. In the present work, all videos annotated with any symbol but X are considered near-duplicates. The dataset contains a total of 13,129 videos consisting of 397,965 keyframes.

In addition, we use the FIVR-200K [26] dataset for validating the results on a second independent dataset. It consists of 225,960 videos collected based on the 4,687 events, and contains 100 video queries. Table 4.5 depicts the annotation labels used in the dataset and their definitions. FIVR-200K includes three different tasks: a) the Duplicate Scene Video Retrieval (DSVR) task where only videos annotated with ND and DS are considered relevant, b) the Complementary Scene Video

Retrieval (CSVR) task which accepts only the videos annotated with ND, DS or CS as relevant, and c) Incident Scene Video Retrieval (ISVR) task where all labels (with the exception of DI) are considered relevant.

(a) CC_WEB_VIDEO

Label	**Transformation**
E	Exactly duplicate
S	Similar video
V	Different version
M	Major change
L	Long version
X	Dissimilar video

(b) FIVR-200K

Label	**Definition**
ND	Near-duplicate
DS	Duplicate scene
CS	Complementary scene
IS	Incident scene
DI	Distractor

Table 4.5: Annotation labels of CC_WEB_VIDEO and FIVR-200K datasets.

4.4.1.2 Development dataset

For generating triplets to train the supervised DML approach, we leverage the VCDB dataset [21]. This dataset is composed of videos from popular video platforms (YouTube and Metacafe) and has been compiled and annotated as a benchmark for the partial copy detection task, which is highly related to the NDVR problem setting. VCDB contains two subsets, the core $\mathscr{C}_c$ and the distractor subset $\mathscr{C}_d$. Subset $\mathscr{C}_c$ contains discrete sets of videos composed by 528 query videos and over 9,000 pairs of partial copies. Each video set has been annotated and the chunks of the video copies extracted. Subset $\mathscr{C}_d$ is a corpus of approximately 100,000 distractor videos that is used to make the video copy detection problem more challenging.

For the triplet generation, we retrieve all video pairs annotated as partial copies. We define an overlap criterion to decide whether to use a pair for the triplet generation: if the duration of the overlap content is greater than a certain threshold t compared to the total duration of each video, then the pair is retained, otherwise discarded. Each video of a given pair can be used once as query and once as positive video. Therefore, the set of query-positive pairs $\mathscr{P}$ is generated based on:

$$\mathscr{P} = \{(q,p) \cup (p,q) | q,p \in \mathscr{C}_c, \mathrm{o}(q,p) > t\} \tag{4.14}$$

where $\mathrm{o}(\cdot,\cdot)$ determines the video overlap. We found empirically that the selection of the threshold t has considerable impact on the quality of the resulting DML model. Subset $\mathscr{C}_d$ is used as the set $\mathscr{N}$ of negatives. To generate hard triplets, the negative videos are selected based on Equation 4.13.

4.4.1.3 Evaluation metrics

To evaluate retrieval performance, we build upon the evaluation scheme described in [55]. We first employ the interpolated *Precision-Recall* (PR) curve. *Precision* is determined as the fraction of retrieved videos that are relevant to the query, and *Recall* as the fraction of the total relevant videos that are retrieved:

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN} \tag{4.15}$$

where TP, FP and FN are the true positives (correctly retrieved), false positives (incorrectly retrieved) and false negatives (missed matches) respectively. The interpolated PR-curve derives from the averaging of the Precision scores over all queries for given Recall ranges. The maximum Precision score is selected as the representative value for each Recall range. We further use *mean Average Precision* (mAP) as defined in [55] to evaluate the quality of video ranking. For each query, the *Average Precision* (AP) is calculated based on:

$$AP = \frac{1}{n}\sum_{i=0}^{n}\frac{i}{r_i} \tag{4.16}$$

where n is the number of relevant videos to the query video, and r_i is the rank of the i-th retrieved relevant video. The mAP is computed from the averaging of the AP across all queries.

4.4.1.4 Implementation details

We experiment with three deep network architectures: AlexNet [30], VGGNet [43] and GoogLeNet [47]. The AlexNet is an 8-layer network that consists of five convolutional/pooling layers, two fully-connected layers and one softmax layer. VGGNet has the same number of fully-connected layers, although the number of convolutional layers may vary. In this paper, the version with 16-layers is employed as it gives similar performance to the 19-layer version. Finally, GoogLeNet is composed of 22 layers in total. In this architecture, multiple convolutions are combined in an intersection module called “inception”. There are nine inception modules in total that are sequentially connected, followed by an average pooling and a softmax layer at the end. All three architectures receive as input images of size 224×224. For all the experiments, the input frames are resized to fit these dimensions, even though this step is not mandatory. Table 4.6 depicts the employed CNN architectures and the number of channels in the respective convolutional layers.

For feature extraction, we use the Caffe framework [19], which provides pre-trained models on ImageNet for both employed CNN networks[5]. Regarding the unsupervised approach, the visual codebooks are generated based on scalable K-

[5] `https://github.com/BVLC/caffe/wiki/Model-Zoo`

(a) AlexNet

Layer $\mathscr{L}^l$	c^l-dim
`conv1`	96
`conv2`	256
`conv3`	384
`conv4`	384
`conv5`	256
total	1376

(b) VGGNet

Layer $\mathscr{L}^l$	c^l-dim
`conv2_1`	128
`conv2_2`	128
`conv3_1`	256
`conv3_2`	256
`conv3_3`	256
`conv4_1`	512
`conv4_2`	512
`conv4_3`	512
`conv5_1`	512
`conv5_2`	512
`conv5_3`	512
total	4096

(c) GoogleNet

Layer $\mathscr{L}^l$	c^l-dim
`inception_3a`	256
`inception_3b`	480
`inception_4a`	512
`inception_4b`	512
`inception_4c`	512
`inception_4d`	528
`inception_4e`	832
`inception_5a`	832
`inception_5b`	1024
total	5488

Table 4.6: Deep CNN architectures and total number of channels per layer used in the proposed approach.

Means++ [2] – the Apache Spark[6] implementation of the algorithm is used for efficiency and scalability – in both aggregation schemes a sample of 100K randomly selected video frames are used for training. The implementation of the supervised deep model is built on Theano [49]. We use [800, 400, 250], [2000, 1000, 500] and [2500, 1000, 500] neurons for the three hidden layers for AlexNet, VGGNet and GoogleNet respectively. Adam optimization [25] is employed with learning rate $l = 10^{-5}$. For the triplet generation, we set $t = 0.8$ which generates approximately 2k pairs in $\mathscr{P}$ and 7M, 4M and 5M triplets in $\mathscr{T}$, for AlexNet, VGGNet and GoogleNet, respectively. Other parameters are set to $\gamma = 1$ and $\lambda = 10^{-5}$.

4.4.1.5 State-of-the-art approaches

We compare the proposed approach with five widely used content-based NDVR approaches. Three of those were developed based on frames of videos sampled from the evaluation set. These are the following:

Auto Color Correlograms (ACC) - Cai et al. [6] use uniform sampling to extract one frame per second for the input video. The auto-color correlograms [14] of each frame are computed and aggregated based on a visual codebook generated from a training set of video frames. The retrieval of near-duplicate videos is performed using tf-idf weighted cosine similarity over the visual word histograms of a query and a dataset video.

Pattern-based approach (PPT) - Chou et al. [9] build a pattern-based indexing tree (PI-tree) based on a sequence of symbols encoded from keyframes, which facilitates the efficient retrieval of candidate videos. They use m-pattern-based dynamic

[6] `http://spark.apache.org`

programming (mPDP) and time-shift m-pattern similarity (TPS) to determine video similarity.

Stochastic Multi-view Hashing (SMVH) - Hao et al. [12] combine multiple keyframe features to learn a group of mapping functions that project video keyframes into the Hamming space. The combination of keyframe hash codes generates a video signature that constitutes the final video representation. A composite Kullback-Leibler (KL) divergence measure is used to compute similarity scores.

The remaining two, which are based on the work of Wu et al. [55], are not built based on any development dataset:

Color Histograms (CH) - This is a global video representation based on the color histograms of keyframes. The color histogram is a concatenation of 18 bins for Hue, 3 bins for Saturation, and 3 bins for Value, resulting in a 24-dimensional vector representation for every keyframe. The global video signature is the normalized color histogram over all keyframes in the video.

Local Structure (LS) - Global signatures and local features are combined using a hierarchical approach. Color signatures are employed to detect near-duplicate videos with high confidence and to filter out very dissimilar videos. For the reduced set of candidate videos, a local feature based method was developed, which compares the keyframes in a sliding window using their local features (PCA-SIFT [24]).

4.4.2 Experimental results

4.4.2.1 Comparison of global feature descriptors

In this section, we benchmark the proposed intermediate CNN features with a number of global frame descriptors used in NDVR literature. The compared descriptors are divided in two groups: handcrafted and learned features[7]. The handcrafted features include RGB histograms, HSV histograms, Local Binary Patterns (LBP), Auto Colour Correlograms (ACC) and Histogram of Oriented Gradients (HOG). For the learned features, we extract the intermediate CNN features, as described in Section 4.3.1.1, and concatenate the layer vectors to generate a single descriptor. Additionally, we experiment with the global features derived from the activations of the first fully connected layer after the convolutional layers, for each architecture. To compare the NDVR performance, a standard bag-of-word scheme with vector aggregation (Section 4.3.1.2) is built based on each global feature descriptor.

Table 4.7 presents the mAP of each model built on a different global descriptor for two different values of K. The intermediate features of GoogleNet and VGGNet achieved the best results with 0.958 and 0.886 for $K = 1,000$ and $K = 10,000$, respectively. In general, learned features lead to considerably better performance than handcrafted ones in both setups. Furthermore, intermediate CNN features outper-

[7] The features have been learned on the ImageNet dataset, since pre-trained networks are utilized. However, ImageNet is a comprehensive dataset, so the learned features can be used in other computer vision tasks (i.e. NDVR) without the need of retraining.

Descriptor/ Network	**layers**	**dimensions**	**K** 1,000	10,000
RGB	-	64	0.857	0.813
HSV	-	162	0.902	0.792
LBP	-	256	0.803	0.683
ACC	-	256	0.936	0.826
HOG	-	1764	**0.940**	**0.831**
AlexNet	conv	1376	0.951	0.879
	fc	4096	0.953	0.875
VGGNet	conv	4096	0.937	**0.886**
	fc	4096	0.936	0.854
GoogleNet	inc	5488	**0.958**	0.857
	fc	1000	0.941	0.849

Table 4.7: mAP and dimensionality of 11 global frame descriptors.

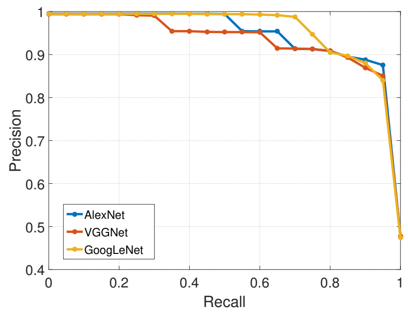
formed the ones derived from the fully connected layers in almost all cases. One may notice that there is a correlation between the dimensions of the descriptors and the performance of the model. Hence, due to the considerable performance difference, we focused our research on the exploration of the potential of intermediate CNN features.

4.4.2.2 Evaluation of BoW approach

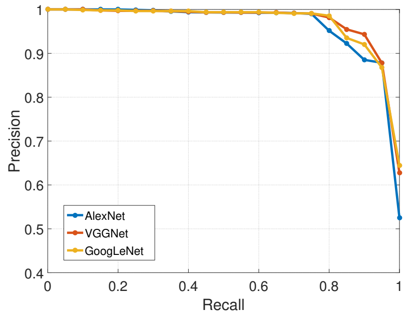
In this section, we study the impact of the feature aggregation scheme, the underlying CNN architecture and the size of the visual vocabulary on the BoW approach. Regarding the first aspect, we benchmark the three CNN architectures with both aggregation schemes using $K = 1,000$ words.

Figure 4.6 depicts the PR curves of the different CNN architectures with the two aggregation schemes. For every CNN architecture, layer-based aggregation schemes outperform vector-based ones. GoogleNet achieves the best vector-based results with a precision close to 100% up to a 70% recall. In terms of recall, all three architectures have similar results in the value range 80%-100%. All three benchmarked architectures have almost perfect performance up to 75% recall when the layer-based aggregation scheme is employed.

As presented in Table 4.8, similar results are obtained in terms of mAP for the CNN architectures and the aggregation schemes. In the case of vector-based aggregation, the results are the same as in Table 4.7, hence GoogleNet outperforms the other two architectures with a mAP of 0.958, and VGGNet reports the worst performance with 0.937 mAP. However, when the layer-based aggregation is employed, VGGNet achieves the best results with a mAP score of 0.976. The lowest, yet competitive results in the case of layer-based aggregation, are obtained for AlexNet with 0.969 mAP.



(a) Vector aggregation



(b) Layer aggregation

Fig. 4.5: Precision-Recall curve of the proposed approach based on three CNN architectures and for the two aggregation schemes.

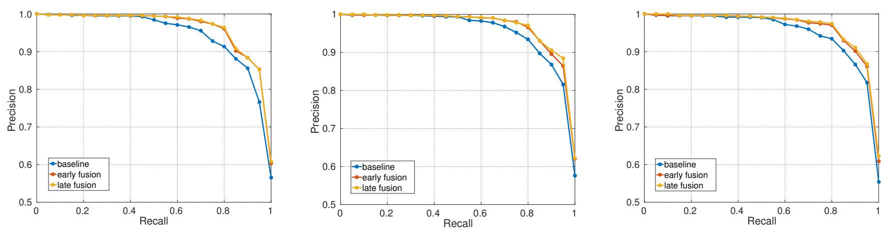
Method	K	AlexNet	VGGNet	GoogleNet
Vector aggr.	1000	0.951	0.937	**0.958**
	10,000	0.879	0.886	0.857
Layer aggr.	1000	0.969	**0.976**	0.974
	10,000	0.948	0.959	0.958

Table 4.8: mAP per CNN architecture and aggregation scheme.

The two schemes are compared with $K = 1,000$ and $K = 10,000$ (Table 4.9) in order to test the impact of vocabulary size. Results reveal that the performance of vector-based aggregation for $K = 10,000$ is lower compared to the case when $K = 1,000$ words are used. It appears that the vector-based aggregation suffers considerably more from the increase of K compared to the layer-based aggregation, which appears to be less sensitive to this parameter. Due to this fact, we did not consider to use the same amount of visual words for the vector-based and the layer-based aggregation, since the performance gap between the two types of aggregation with the same number of visual words would be much more pronounced.

4.4.2.3 Evaluation of DML approach

In this section, we study the performance of the supervised DML approach in the evaluation dataset in relation to the underlying CNN architecture and the different fusion schemes. The three CNN architectures are benchmarked. For each of them, three configurations are tested: i) *baseline*: all frame descriptors are averaged to a single vector which is used for retrieval without any transformation, ii) *early fusion*: all frame descriptors are averaged to a single vector which is then transformed by applying the learned embedding function to generate the video descriptor, iii) *late*



(a) AlexNet (b) VGGNet (c) GoogleNet

Fig. 4.6: Precision-Recall curve of the baseline and two DML fusion schemes for the three benchmarked CNN architectures.

fusion: all frame descriptors are transformed by applying the learned embedding function and the generated embeddings are then averaged.

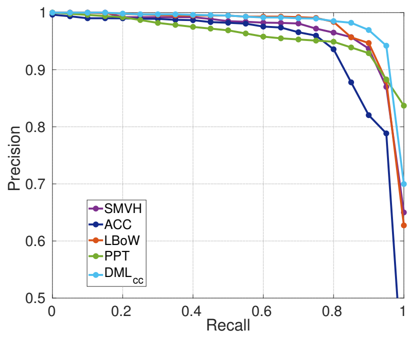
Architecture	**baseline**	**early fusion**	**late fusion**
AlexNet	0.948	0.964	0.964
VGGNet	0.956	0.970	**0.971**
GoogleNet	0.952	0.968	0.969

Table 4.9: mAP of the baseline and two DML fusion schemes for the three benchmarked CNN architectures.

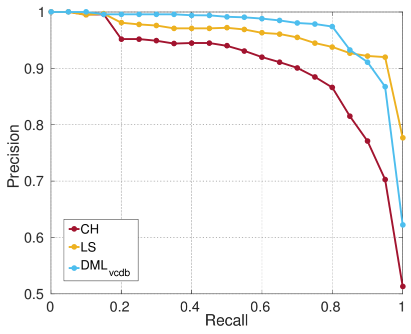
Figure 4.6 and Table 4.9 presents the PR curves and the mAP, respectively, of the three CNN architectures with the three fusion setups. Late fusion schemes consistently outperform the other two fusion schemes for all CNN architectures. VGGNet achieves the best results for all three settings with a small margin compared to the GoogleNet, with precision more than 97% up to 80% recall and mAP scores of 0.970 and 0.971 for early and late fusion respectively. Performance clearly increases in both trained fusion schemes compared to the baseline for all three architectures. The early and late fusion schemes achieve almost identical results, which is an indication that the choice of the fusion scheme is not critical.

4.4.2.4 Comparison against state-of-the-art NDVR approaches

For comparing the performance of the two approaches with the five NDVR approaches from the literature, we select the setup using VGGNet features with layer aggregation for the BoW approach, denoted as LBoW, and the setup using VGGNet features with late fusion for the DML approach, denoted as DML$_{vcdb}$, since they achieved the best results in each case. We separate the compared approaches in two groups based on the developed dataset, i.e. whether the evaluation dataset is used



(a) Same dataset



(b) No/Different dataset

Fig. 4.7: Precision-Recall curve comparison between the two developed approaches against five state-of-the-art methods. The approaches are divided based on the dataset used for development.

for development or not. For the sake of comparison and completeness, the results of the DML method trained on a triplet set derived from both VCDB (similar to DML$_{vcdb}$) and also videos sampled from CC_WEB_VIDEO are denoted as DML$_{cc}$. This simulates the situation where the DML-based approach has access to a portion of the evaluation corpus, similar to the setting used by the competing approaches.

In Table 4.10, the mAP scores of the competing methods are reported. The DML approach outperforms all methods in each group with a clear margin. A similar conclusion is reached from comparing the PR curves illustrated in Fig. 4.7, with the light blue line (DML approach) lying upon all others up to 90% recall in both cases. The DML approach trained on VCDB dataset outperforms four out of five state-of-the-art methods. It achieves similar results to the SMVH, even though the latter has been developed with access to the evaluation dataset during training. The LBoW approach is in the second place consistently outperforming all five competing approaches by a considerable margin.

	Same dataset					No/Different dataset		
Method	**ACC**	**PPT**	**SMVH**	**LBoW**	**DML$_{cc}$**	**CH**	**LS**	**DML$_{vcdb}$**
mAP	0.944	0.958	0.971	0.976	**0.982**	0.892	0.954	**0.971**

Table 4.10: mAP comparison between the two developed approaches against five state-of-the-art methods. The approaches are divided based on the dataset used for development.

4.4.2.5 In-depth comparison of the two approaches

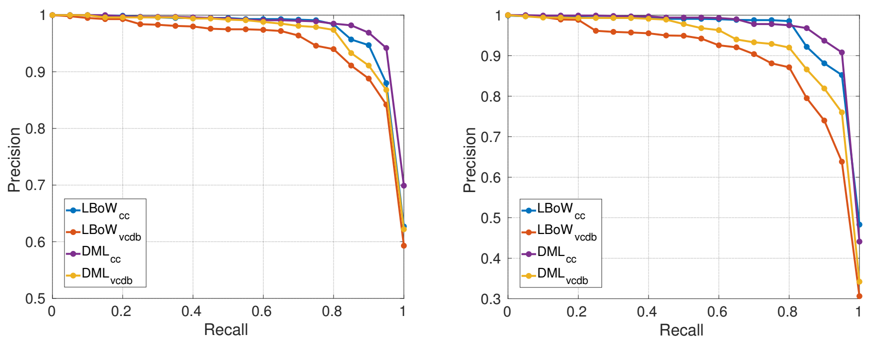
In this section, we compare the two implemented NDVR approaches in two evaluation settings. To this end, in addition to the existing experiments, we implement the BoW approach with VGGNet features and layer aggregation based on information derived from the VCDB dataset, i.e. we build the layer codebooks from a set of video frames sampled from the aforementioned dataset. We then test two variations, the LBoW$_{cc}$ that was developed on the CC_WEB_VIDEO dataset (same as Section 4.4.2.2) and the LBoW$_{vcdb}$ developed on the VCDB dataset. For each of the 24 queries of CC_WEB_VIDEO, only the videos contained in its subset (the dataset is organized in 24 subsets, one per query) are considered as candidate and used for the calculation of retrieval performance. To emulate a more challenging setting, we created CC_WEB_VIDEO* in the following way: for every query in CC_WEB_VIDEO, the set of candidate videos is the entire dataset instead of only the query subset.

	CC_WEB_VIDEO	CC_WEB_VIDEO*
LBoW$_{vcdb}$	0.957	0.906
DML$_{vcdb}$	**0.971**	**0.936**
LBoW$_{cc}$	0.976	0.960
DML$_{cc}$	**0.982**	**0.969**

Table 4.11: mAP comparison of the two developed approaches on two different dataset setups.

Figure 4.8 depicts the PR curves of the four runs and the two setups. There is a clear difference between the performance of the two variants of the LBoW approach, for both dataset setups. The DML approach outperforms the LBoW approach for all runs and setups at any recall point by a large margin. Similar conclusions can be drawn from the mAP scores of Table 4.11. The performance of LBoW drops by more than 0.02 and 0.062 when the codebook is learned on VCDB, for each setup respectively. Again, there is a considerable drop in performance in CC_WEB_VIDEO* setup for both approaches, with the DML being more resilient to the setup change. As a result, it has been demonstrated to be highly competitive and possible to transfer to different datasets with relatively lower performance loss.

In addition, the developed approaches are also benchmarked on the FIVR-200K [26] dataset. As described in Section 4.2.3 it includes three tasks that accept different type of video results as relevant. To compare the two methods, we implemented them with frame features derived from the VGGNet and built them with videos from the VCDB dataset. Table 4.12 present the mAP of the two developed approaches on the FIVR-200K dataset. It is evident that the DML approach achieves noticeably better performance in comparison to the LBoW, when they are both developed on a different dataset other than the evaluation. For the DSVR task, the two methods achieve 0.4 and 0.378 mAP for DML and LBoW, respectively. The performance of



(a) CC_WEB_VIDEO (b) CC_WEB_VIDEO*

Fig. 4.8: Precision-Recall curve comparison of the two developed approaches on two dataset setups.

both approaches marginally drops for the CSVR task in comparison to DSVR with a reduction of about 0.02 in terms of mAP. On the ISVR task, both runs have a considerable drop in their performance, with 0.312 and 0.297 mAP for DML and LBoW, respectively. Hence, the performance of both methods is significantly reduced in comparison to CC_WEB_VIDEO dataset, revealing that the FIVR-200K dataset is much more challenging. The main reason is that the vast majority of positive video pairs are partially related, i.e. the videos are not related in their entirety but in particular segments. The competing approaches from the NDVR literature lead to even lower performance, since they are based on video-level schemes that employ handcrafted frame descriptors with limited representation capability.

task	**DSVR**	**CSVR**	**ISVR**
LBoW	0.378	0.361	0.297
DML	**0.398**	**0.378**	**0.309**

Table 4.12: mAP of the two developed approaches on the FIVR-200K dataset.

Both presented approaches are limited in similar ways which leads to similar errors in the retrieval process. The major issue of both approaches is that they do not function effectively when the near-duplicate segment between two videos is small relative to their total size. As revealed from the evaluation in FIVR-200K dataset, video-level solutions suffer in such setups. Even the LBoW approach where the video-level representation contains frame-level information fails to retrieve relevant videos, especially when it has been built on different dataset than the evaluation. Another category of videos that the proposed schemes fail is when heavy transformations have been applied on the source video. Typically, the extracted frame descriptors are not close enough, so as such videos to be retrieved and ranked with high similarity score. Even the DML scheme that should learn to handle such case fails to recognize this kind of duplicate pairs, especially when heavy edits or overlays have been applied. A solution to this issue is the use of frame descriptors that better capture local information within frames. This can be achieved with end-to-end training of the CNN models and/or use of another aggregation function (other than MAC) that better preserves local information.

Finally, we compare the two approaches in terms of processing time on the large-scale FIVR-200K dataset. The results have been measured using the open source library Scikit-learn [53] in Python, on a Linux PC with a 4-core i7-4770K and 32GB of RAM. The DML approach is significantly faster than the LBoW approach. It needs 333 ms to perform retrieval for one query on FIVR-200K dataset, compared to 1,155 ms needed for the LBoW approach. However, both methods are significantly faster than common frame-level approaches, which usually need several minutes to process all videos in the dataset.

4.5 Conclusions and future work

In this chapter, we focused on the problem of Near-Duplicate Video Retrieval (NDVR). First we presented a review of NDVR definitions, approaches and datasets existing in the literature. The state-of-the art methods were grouped in three major categories based on the level of video matching they perform: video-level, frame-level and filter-and-refine matching. Moreover, we proposed two different video-level approaches (an unsupervised and a supervised) based on deep neural networks. For both methods, we used CNN features extracted from the intermediate convolutional layers by applying Maximum Activations of Convolutions (MAC). We found that this setup led to the best results among many other features, both hand-crafted and learned.

The first approach is an unsupervised scheme that relies on a Bag-of-Word (BoW) video representation. A layer-based aggregation scheme was introduced in order to generate the global video representation, and then store it in an inverted file index for fast indexing and retrieval. To quantify video similarity, we calculated the cosine similarity on *tf-idf* weighted versions of the extracted vectors and ranked the results in descending order. However, we found that there are several limitations regarding the BoW approach, i.e. it is a dataset-specific solution and is hard to be re-trained on new data. To address these issues, we developed a second supervised approach based on DML. This method approximates an embedding function that transforms input frame descriptors and leads to more accurate computation of the distance between two candidate videos. For each video in the dataset, we sampled one frame per second and extracted its CNN features to generate a global video vector. The global video vectors are then transformed based on the embedding function to the learned feature space. The video retrieval is performed based on the Euclidean distance of the video embeddings.

We conducted extensive evaluations with different experimental setups, testing the performance of the developed approaches under various settings. Through the evaluation process, it was evident that the developed approaches exceed the performance of five established state-of-the-art NDVR approaches. Finally, we empirically determined that the DML approach overcomes the limitations imposed by the BoW approach, i.e. it achieves better performance even without access to the evaluation dataset (even though further improvements are possible if such access is possible).

In the future, we will focus on the improvement of the retrieval performance of the developed system. Initially, we are going to put effort on the design of sophisticated similarity calculation functions that take into account the spatial structure of video frames and, at the same time, the temporal relations within frame sequences, in order to precisely compute the similarity between two compared videos. To achieve these goals, we will modify the developed DML approach to perform frame-level matching, e.g. by considering more effective fusion schemes (compared to early and late fusion), so as to capture the temporal relations between videos. To capture the spatial structure of video frames during the similarity calculation process, we are going to devise a solution that computes similarity at region level. Moreover, we plan to exploit the spatio-temporal information contained in consecutive video frames by employing 3D and/or two-steam CNN network architectures to extract video features. These networks are able to encode the depicted actions in videos to compact feature representations. We anticipate that such features will have considerable impact on the performance of the systems, especially in more general retrieval tasks such as ISVR. Finally, we will assess the performance of the developed approach on the problem of Partial Duplicate Video Retrieval (PDVR).

References

1. TREC Video Retrieval Evaluation: TRECVID (2018). URL `https://trecvid.nist.gov/`
2. Bahmani, B., Moseley, B., Vattani, A., Kumar, R., Vassilvitskii, S.: Scalable k-means++. Proceedings of the VLDB Endowment **5**(7), 622–633 (2012)
3. Baraldi, L., Douze, M., Cucchiara, R., Jégou, H.: LAMV: Learning to align and match videos with kernelized temporal layers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7804–7813 (2018)
4. Basharat, A., Zhai, Y., Shah, M.: Content based video matching using spatiotemporal volumes. Computer Vision and Image Understanding **110**(3), 360–377 (2008)
5. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision, pp. 404–417. Springer (2006)
6. Cai, Y., Yang, L., Ping, W., Wang, F., Mei, T., Hua, X.S., Li, S.: Million-scale near-duplicate video retrieval system. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 837–838. ACM (2011)
7. Chen, J., Huang, T.: A robust feature extraction algorithm for audio fingerprinting. In: Pacific-Rim Conference on Multimedia, pp. 887–890. Springer (2008)
8. Cherubini, M., De Oliveira, R., Oliver, N.: Understanding near-duplicate videos: a user-centric approach. In: Proceedings of the 17th ACM international conference on Multimedia, pp. 35–44. ACM (2009)
9. Chou, C.L., Chen, H.T., Lee, S.Y.: Pattern-based near-duplicate video retrieval and localization on web-scale videos. IEEE Transactions on Multimedia **17**(3), 382–395 (2015)
10. Douze, M., Jégou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-filtering. IEEE Transactions on Multimedia **12**(4), 257–266 (2010)
11. Hao, Y., Mu, T., Goulermas, J.Y., Jiang, J., Hong, R., Wang, M.: Unsupervised t-distributed video hashing and its deep hashing extension. IEEE Transactions on Image Processing **26**(11), 5531–5544 (2017)
12. Hao, Y., Mu, T., Hong, R., Wang, M., An, N., Goulermas, J.Y.: Stochastic multiview hashing for large-scale near-duplicate video retrieval. IEEE Transactions on Multimedia **19**(1), 1–14 (2017)
13. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern recognition **42**(3), 425–436 (2009)
14. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pp. 762–768. IEEE (1997)
15. Huang, Z., Shen, H.T., Shao, J., Cui, B., Zhou, X.: Practical online near-duplicate subsequence detection for continuous video streams. IEEE Transactions on Multimedia **12**(5), 386–398 (2010)
16. Huang, Z., Shen, H.T., Shao, J., Zhou, X., Cui, B.: Bounded coordinate system indexing for real-time video clip search. ACM Transactions on Information Systems (TOIS) **27**(3), 17 (2009)
17. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European conference on computer vision, pp. 304–317. Springer (2008)
18. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3304–3311. IEEE (2010)
19. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, pp. 675–678. ACM (2014)
20. Jiang, M., Tian, Y., Huang, T.: Video copy detection using a soft cascade of multimodal features. In: 2012 IEEE International Conference on Multimedia and Expo, pp. 374–379. IEEE (2012)

21. Jiang, Y.G., Jiang, Y., Wang, J.: VCDB: A Large-Scale Database for Partial Copy Detection in Videos. In: European Conference on Computer Vision, pp. 357–371. Springer (2014)
22. Jiang, Y.G., Wang, J.: Partial copy detection in videos: A benchmark and an evaluation of popular methods. IEEE Transactions on Big Data **2**(1), 32–42 (2016)
23. Jing, W., Nie, X., Cui, C., Xi, X., Yang, G., Yin, Y.: Global-view hashing: harnessing global relations in near-duplicate video retrieval. World Wide Web pp. 1–19 (2018)
24. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, pp. II–II. IEEE
25. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, I.: FIVR: Fine-grained incident video retrieval. IEEE Transactions on Multimedia (2019)
27. Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, Y.: Near-Duplicate Video Retrieval by Aggregating Intermediate CNN Layers. In: International Conference on Multimedia Modeling, pp. 251–263. Springer (2017)
28. Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, Y.: Near-Duplicate Video Retrieval with Deep Metric Learning. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 347–356. IEEE (2017)
29. Kraaij, W., Awad, G.: TRECVID 2011 content-based copy detection: Task overview. Online Proceedings of TRECVid 2010 (2011)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
31. Law-To, J., Joly, A., Boujemaa, N.: Muscle-VCD-2007: a live benchmark for video copy detection (2007)
32. Liu, H., Lu, H., Xue, X.: A segmentation and graph-based video sequence matching method for video copy detection. IEEE Transactions on knowledge and data engineering **25**(8), 1706–1718 (2013)
33. Liu, J., Huang, Z., Cai, H., Shen, H.T., Ngo, C.W., Wang, W.: Near-duplicate video retrieval: Current research and future trends. ACM Computing Surveys (CSUR) **45**(4), 44 (2013)
34. Liu, L., Lai, W., Hua, X.S., Yang, S.Q.: Video histogram: A novel video signature for efficient web video duplicate detection. In: International Conference on Multimedia Modeling, pp. 94–103. Springer (2007)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
36. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International journal of computer vision **60**(1), 63–86 (2004)
37. Poullot, S., Tsukatani, S., Phuong Nguyen, A., Jégou, H., Satoh, S.: Temporal matching kernel with explicit feature maps. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 381–390. ACM (2015)
38. Revaud, J., Douze, M., Schmid, C., Jégou, H.: Event retrieval in large video collections with circulant temporal encoding. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 2459–2466. IEEE (2013)
39. Shang, L., Yang, L., Wang, F., Chan, K.P., Hua, X.S.: Real-time large scale near-duplicate web video retrieval. In: Proceedings of the 18th ACM international conference on Multimedia, pp. 531–540. ACM (2010)
40. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN Features off-the-shelf: an Astounding Baseline for Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
41. Shen, H.T., Zhou, X., Huang, Z., Shao, J., Zhou, X.: UQLIPS: A Real-time Near-duplicate Video Clip Detection System. In: Proceedings of the 33rd international conference on Very large data bases, pp. 1374–1377. VLDB Endowment (2007)
42. Silverman, C.: Verification handbook (2013)

43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
44. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pp. 1470–1477. IEEE (2003)
45. Song, J., Yang, Y., Huang, Z., Shen, H.T., Hong, R.: Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 423–432. ACM (2011)
46. Song, J., Yang, Y., Huang, Z., Shen, H.T., Luo, J.: Effective multiple feature hashing for large-scale near-duplicate video retrieval. IEEE Transactions on Multimedia **15**(8), 1997–2008 (2013)
47. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
48. Tan, H.K., Ngo, C.W., Hong, R., Chua, T.S.: Scalable detection of partial near-duplicate videos by visual-temporal consistency. In: Proceedings of the 17th ACM international conference on Multimedia, pp. 145–154. ACM (2009)
49. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints **abs/1605.02688** (2016)
50. Tian, Y., Huang, T., Jiang, M., Gao, W.: Video copy-detection and localization with a scalable cascading framework. IEEE MultiMedia **20**(3), 72–86 (2013)
51. Tian, Y., Jiang, M., Mou, L., Rang, X., Huang, T.: A multimodal video copy detection approach with sequential pyramid matching. In: 2011 18th IEEE International Conference on Image Processing, pp. 3629–3632. IEEE (2011)
52. Tian, Y., Qian, M., Huang, T.: Tasc: A transformation-aware soft cascading approach for multimodal video copy detection. ACM Transactions on Information Systems (TOIS) **33**(2), 7 (2015)
53. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in python. PeerJ **2**, e453 (2014)
54. Wang, L., Bao, Y., Li, H., Fan, X., Luo, Z.: Compact CNN Based Video Representation for Efficient Video Copy Detection. In: International Conference on Multimedia Modeling, pp. 576–587. Springer (2017)
55. Wu, X., Hauptmann, A.G., Ngo, C.W.: Practical elimination of near-duplicates from web video search. In: Proceedings of the 15th ACM international conference on Multimedia, pp. 218–227. ACM (2007)
56. Wu, Z., Aizawa, K.: Self-similarity-based partial near-duplicate video retrieval and alignment. International Journal of Multimedia Information Retrieval **3**(1), 1–14 (2014)
57. Yang, Y., Tian, Y., Huang, T.: Multiscale video sequence matching for near-duplicate detection and retrieval. Multimedia Tools and Applications pp. 1–26 (2018)
58. Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 53–61 (2015)
59. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE transactions on pattern analysis and machine intelligence **29**(6), 915–928 (2007)
60. Zheng, L., Zhao, Y., Wang, S., Wang, J., Tian, Q.: Good practice in cnn feature transfer. arXiv preprint arXiv:1604.00133 (2016)