Near-Duplicate Video Retrieval with Deep Metric Learning

Giorgos Kordopatis-Zilos[1,2] Symeon Papadopoulos[1] Ioannis Patras[2] Yiannis Kompatsiaris[1]

[1]Information Technologies Institute, CERTH, Thessaloniki, Greece
[2]Queen Mary University of London, Mile End road, E1 4NS London, UK
{georgekordopatis,papadop,ikom}@iti.gr i.patras@qmul.ac.uk

Abstract

This work addresses the problem of Near-Duplicate Video Retrieval (NDVR). We propose an effective video-level NDVR scheme based on deep metric learning that leverages Convolutional Neural Network (CNN) features from intermediate layers to generate discriminative global video representations in tandem with a Deep Metric Learning (DML) framework with two fusion variations, trained to approximate an embedding function for accurate distance calculation between two near-duplicate videos. In contrast to most state-of-the-art methods, which exploit information deriving from the same source of data for both development and evaluation (which usually results to dataset-specific solutions), the proposed model is fed during training with sampled triplets generated from an independent dataset and is thoroughly tested on the widely used CC_WEB_VIDEO dataset, using two popular deep CNN architectures (AlexNet, GoogleNet). We demonstrate that the proposed approach achieves outstanding performance against the state-of-the-art, either with or without access to the evaluation dataset.

1. Introduction

Near-duplicate video retrieval (NDVR) is a research topic of increasing interest in recent years, due to the exponential growth of social media applications and video sharing websites, which typically feature vast amounts of near-duplicate content. The problem is exacerbated in the case of video due to its considerably larger volume (compared to text and images), which make it a great challenge for every web-based video platform as well as for systems that analyze and index large amounts of web video content. As a result, efficient retrieval of near-duplicate videos is nowadays an indispensable component in numerous applications including video search, management, recommendation, copy detection and copyright protection.

The definition of near-duplicate videos (NDVs) is a controversial topic in the multimedia research community, with several definitions proposed that differ with respect to the required level of similarity between NDVs [17]. In this work, we adopt the definition from Wu *et al*. [31], where NDVs are defined as videos that are close to duplicate of each other, but different in terms of photometric variations (color, lighting changes), editing operations (caption, logo and border insertion), encoding parameters, file format, different lengths, and other modifications. A number of NDV examples are illustrated in Figure 1.

Considerable effort has been invested by the research community on the problem of NDVR. However, many state-of-the-art methods adopt a dataset-bound approach and use the same dataset for both development and evaluation. This leads to specialized solutions that typically exhibit poor performance when used (without tuning) on different video corpora. For instance, some methods learn codebooks [24, 1, 4, 14] or hashing functions [25, 26, 7] based on sample frames from the evaluation dataset, and as a result their reported retrieval performance is often exaggerated.

Motivated by the excellent performance of deep learning in a wide variety of multimedia problems, we are proposing a video-level NDVR approach that incorporates deep learning in two steps. First, we use CNN features from intermediate convolution layers based on a well-known scheme called Maximum Activation of Convolutions [22, 34, 21], which was recently used for NDVR and led to improved results [14]. Second, we leverage a Deep Metric Learning (DML) framework based on a triplet-wise scheme, which has been shown to be effective in a variety of cases [2, 30, 29]. To our knowledge, it is the first time that deep metric learning is exploited for NDVR. In particular, we train a Deep Neural Network (DNN) to learn an embedding function that maps videos to a feature space where NDVs have smaller distances between each other compared to other videos. Moreover, two different fusion variations are proposed for the generation of video representation. The

Figure 1. Examples of queries and near-duplicate videos from CC_WEB_VIDEO dataset.

generated video representation is compact in order to facilitate the development of scalable NDVR systems.

We also propose a triplet generation method for training the DML framework with video samples from the VCDB [11] dataset. The proposed approach is evaluated on the widely used CC_WEB_VIDEO dataset [31], with CNN features from two popular architectures [16, 27]. To compare with the state of the art, we are also evaluating our approach using training data from the target video corpus, simulating the evaluation setting of competing approaches. Our system outperforms these approaches, with more than 0.007 mAP in all experimental setups.

2. Related Work

A thorough study on the NDVR problem and several recent approaches is provided by Liu *et al*. [17]. According to it, existing NDVR methods are classified based on the granularity of the matching between NDVs into video-, frame- and hybrid-level matching.

Video-level matching: These approaches aim at solving the NDVR problem at massive scale. Videos are usually represented with a global signature such as an aggregate feature vector [31, 18, 9] or a hash code [25, 7, 26] and the video matching is based on the computation of the pairwise similarity between the corresponding video representations.

Frame-level matching: NDVs are determined in this case by comparing between individual frames or frame sequences of the candidate videos. Existing approaches [5, 1, 14] calculate frame-by-frame similarity based on Bag-of-Words (BoW) schemes or employ sequence alignment algorithms. Other works have explored spatio-temporal representations [24, 33] for improving retrieval performance and accelerating the similarity computation.

Hybrid-level matching: Such approaches attempt to combine the advantages of video- and frame-level methods. Typical such approaches are, for instance, presented in [31, 4], both of which first employ a filter-and-refine scheme to cluster and filter out near-duplicate videos, and then use frame-to-frame similarity on the reduced set of videos.

Moreover, the NDVR problem is related to the well-known TRECVID copy detection task [15]. The main difference in the TRECVID copy detection task is that video copies are artificially generated by applying standard transformations to a corpus of videos, whereas in case of NDVR duplicates correspond to actual user submitted videos.

Another field of related work is metric learning, on which a detailed survey is provided by Yang and Jin [32]. Metric learning is conducted using pairwise [6, 35, 19, 21] or triplet-wise constraints [2, 30, 29, 23, 3]. Its main purpose is to learn an optimal projection for mapping input features to another feature space. In the case of NDVR, we aim at an embedding function that maps NDVs closer to each other than to the rest of videos.

Pairwise methods usually employ contrastive loss that tries to minimize the distance between pairs of examples with same-class label, while penalizing examples with different-class labels that are closer than a margin γ [6, 21]. Triplet-wise embedding is trained on triplets of data with an anchor point, a positive that belongs to the same class, and a negative that belongs to a different class [29, 23, 3]. Triplet-wise methods use a loss over triplets to push the anchor and positive close, and penalize triplets where the distance be-

tween anchor and negative is less than the one between anchor and positive plus a margin γ. Deep metric learning has been successfully applied to a variety of problems including image retrieval [30, 29, 21], face recognition/retrieval [23], person re-identification [3, 20], etc.

3. Approach Overview

The proposed NDVR approach leverages features produced by the intermediate convolution layers of deep CNN architectures (section 3.1) to generate compact global video representations. Additionally, to accurately compute the similarity between two candidate videos, a DNN is trained to approximate an embedding function for the distance calculation (section 3.2). The model is built on batches of generated triplets from a development dataset (section 3.3).

3.1. Feature Extraction

We adopt a compact representation to extract frame descriptors that is derived from activations of convolution layers of a pre-trained CNN. This image representation is called Maximum Activation of Convolutions (MAC) [22, 34, 21, 14]. To this end, a pre-trained CNN network is employed, with a total number of L convolution layers, denoted as $\mathcal{L}^1, \mathcal{L}^2, ..., \mathcal{L}^L$. Forward propagating an image through the network generates a total of L feature maps, denoted as $\mathcal{M}^l \in \mathbb{R}^{n_d^l \times n_d^l \times c^l} (l = 1, ..., L)$, where $n_d^l \times n_d^l$ is the dimension of every channel for convolution layer $\mathcal{L}^l$ (which depends on the size of the input image) and c^l is the total number of channels. To extract a single descriptor from every layer, we apply max pooling on every channel of feature map $\mathcal{M}^l$ to extract a single value. The extraction process is formulated in Equation 1.

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

where layer vector v^l is a c^l-dimensional vector that is derived from max pooling on every channel of feature map $\mathcal{M}^l$. After extraction, all layer vectors are concatenated to a single descriptor. Finally, the frame descriptors are normalized by applying zero-mean and ℓ_2-normalization.

We experiment with two deep network architectures: AlexNet [16] and GoogleNet [27]. For the former, all convolution layers are used for the extraction of the frame descriptors, whereas, for the latter, all inception layers. The generated vectors have 1,376 and 5,488 dimensions respectively. Both architectures receive images of size 224×224 as input (input frames are resized to these dimensions).

To generate global video descriptors, uniform sampling is initially applied to select n frames per second for every video (in our setup we use $n = 1$) and extract the respective features for each of them. Global video descriptors 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, because the weights of the pre-trained network that is used for feature extraction are not updated.

3.2. Metric Learning

3.2.1 Problem setting

We address the problem of learning a pairwise similarity function for NDVR from the relative information of pair/triplet-wise video relations. For a given query video and a set of candidate videos, the goal is to compute the similarity between the query and every candidate video and use it for ranking the entire set of candidates in the hope that the NDVs are retrieved at the top ranks. To formulate this process, we define the similarity between two arbitrary videos q and p as the squared Euclidean distance in the video embedding space (Equation 2).

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

where $f_\theta(\cdot)$ is the embedding function that maps a video to a point in an 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 $\mathrm{I}(\cdot, \cdot)$, which specifies whether a pair of videos are near-duplicate.

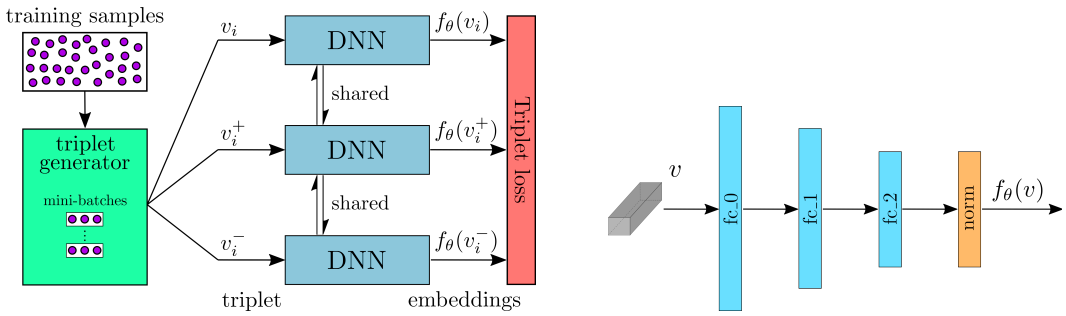
$$\mathrm{I}(q, p) = \begin{cases} 1 & \text{if } q, p \text{ are NDVs} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Our objective is to learn an embedding function $f_\theta(\cdot)$ that assigns smaller distances to NDV pairs compared to non-NDV ones. 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 representations to a common space $\mathbb{R}^d$, where d is the dimension of the feature embedding, in which the distance between query v and positive v^+ is always smaller than the distance between query v and negative v^- (Equation 4).

$$\begin{aligned} &\mathrm{D}(f_\theta(v), f_\theta(v^+)) < \mathrm{D}(f_\theta(v), f_\theta(v^-)), \\ &\forall v, v^+, v^- \text{ such that } \mathrm{I}(v, v^+) = 1, \mathrm{I}(v, v^-) = 0 \end{aligned} \tag{4}$$

3.2.2 Triplet loss

To implement the learning process, we create a collection of N training instances organized in the forms of triplets $\mathcal{T} = \{(v_i, v_i^+, v_i^-), i = 1, ..., N\}$, 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



(a) DML architecture (b) DNN

Figure 2. Illustration of (a) the DML architecture, and (b) the composition of the DNN.

for a given triplet called 'triplet loss' (Equation 5).

$$L_\theta(v_i, v_i^+, v_i^-) = \max\{0, \mathrm{D}(f_\theta(v_i), f_\theta(v_i^+)) - \mathrm{D}(f_\theta(v_i), f_\theta(v_i^-)) + \gamma\} \quad (5)$$

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 described in Equation 6.

$$\min_\theta \sum_{i=1}^{m} L_\theta(v_i, v_i^+, v_i^-) + \lambda \left\|\theta\right\|_2^2 \quad (6)$$

where λ is a regularization parameter to prevent overfitting of the model, 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 in place, the model will eventually learn a video representation that improves the effectiveness of the NDVR solution.

3.2.3 DML architecture

For training the DML model, a triplet-based network architecture is proposed (Figure 2(a)) that optimizes the triplet loss function of Equation 5. The network is provided with a set of triplets $\mathcal{T}$ created by the triplet generation process of section 3.3. 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. $\left\|f_\theta(v)\right\|_2 = 1$ (Figure 2(b)). 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 5.

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. Two variants are proposed for fusing similarity computation across video frames: early and late fusion (Figure 3).

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 $f_\theta(\cdot)$.

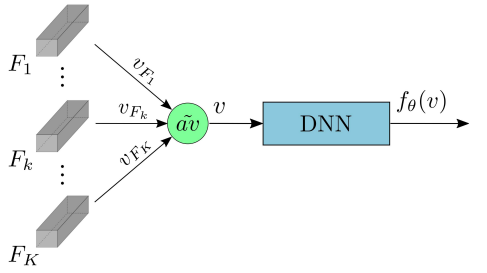
Late fusion: Every extracted frame descriptor of an input video is fed forward to the network, and the set of their embedding transformations is averaged and normalized.

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.

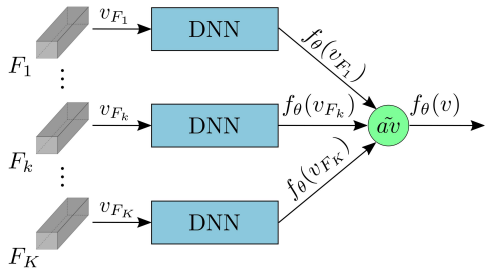
Finally, the similarity between two videos derives from the distance of their representations. For a given query q and a set of M candidate videos $\{p_i\}_{i=1}^M \in P$, the similarity within each candidate pair is determined by Equation 7.

$$\mathrm{S}(q, p) = 1 - \frac{\mathrm{D}(f_\theta(q), f_\theta(p))}{\max\limits_{p_i \in P}(\mathrm{D}(f_\theta(q), f_\theta(p_i)))} \quad (7)$$

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



(a) Early fusion



(b) Late fusion

Figure 3. Illustration of early and late fusion schemes.

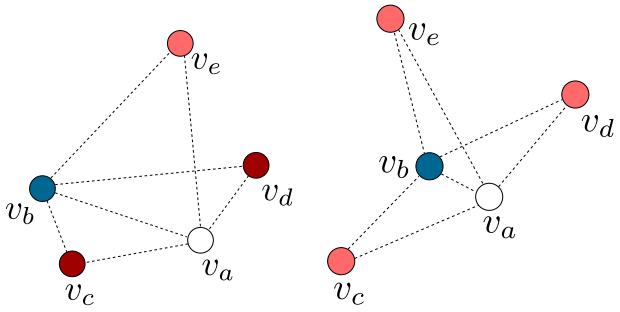
3.3. Triplet Generation

A critical component 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 to the total number of 3-combinations over the size N of the video corpus, i.e. $\binom{N}{3} = \frac{N \cdot (N-1) \cdot (N-2)}{6}$. We have empirically determined that only a tiny portion of videos in a video 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 $N = 1000$, 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: $\mathcal{P}$, a set of near duplicate video pairs that are used as query-positive pairs, and $\mathcal{N}$, 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 Equation 8.

$$\mathcal{T} = \{(q, p, n) | (q, p) \in \mathcal{P}, n \in \mathcal{N}, \mathrm{D}(q, p) > \mathrm{D}(q, n)\} \quad (8)$$

where $\mathcal{T}$ is the resulting set of triplets. The global video features are first extracted following the process of section 3.1. Then, the distance between every query in $\mathcal{P}$ and every dissimilar video in $\mathcal{N}$ is calculated. If the query-positive distance is greater than a query-negative distance, then a hard triplet is formed composed by the three videos. The distance is calculated based on the Euclidean distance of the initial global video descriptors.



(a) Before training (b) After training

Figure 4. Examples of video representations in feature space before and after training. Colours: (white) query video (blue) NDV (red) distractor videos.

Figure 4 visualizes the training and triplet generation process. Figure 4(a) depicts the videos in feature space before training. The white and blue colour circles represent the query and near-duplicate videos, respectively, whereas the dissimilar videos are painted in red colour. In particular, v_a is the query and v_b is a NDV. However, before training, it is clear that their distance D_{ab} is greater than distances D_{ac} and D_{ad}; therefore, v_c and v_d (deep red) are *hard negatives* and two triplets will be created $\{v_a, v_b, v_c\}$ and $\{v_a, v_b, v_d\}$. The video v_e (light red) does not generate any triplet because its distance from the two NDVs is greater than the distance between them. After training, the distance between the query and the NDV must be smaller than their distance to any other dissimilar video, as illustrated in Figure 4(b).

4. Evaluation

4.1. Experimental setup

Development dataset: We leverage the VCDB dataset [11] to generate triplets for training our DML-based system. This dataset is composed of videos derived from popular video platforms (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 $\mathcal{C}_c$ and the distractor subset $\mathcal{C}_d$. Subset $\mathcal{C}_c$ 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 video chunks of the video copies have been extracted. Subset $\mathcal{C}_d$ 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 that have been annotated as partial copies. We define an overlap criterion that determines whether a pair is going to be used 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, it is 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 $\mathcal{P}$ is generated based on Equation 9.

$$\mathcal{P} = \{(q,p) \cup (p,q) | q,p \in \mathcal{C}_c, \mathrm{o}(q,p) > t\} \qquad (9)$$

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 $\mathcal{C}_d$ is used as the set $\mathcal{N}$ of negatives. To generate hard triplets, the negative videos are selected from $\mathcal{C}_d$ based on Equation 8.

Evaluation dataset: Experiments were performed on the CC_WEB_VIDEO dataset [31]. The collection consists of a set of videos retrieved by submitting 24 frequent text queries to popular video sharing websites, i.e. YouTube, Google Video, and Yahoo! Video. The dataset contains a total of 13,129 videos with 397,965 keyframes. In addition to the provided keyframes, we extracted one frame per second for every video in the dataset resulting in a total of approximately 2M video frames. Some of the approaches of section 4.2 rely on the dataset keyframes, while others on the extracted frames.

Evaluation metrics: To measure detection accuracy, we employ the interpolated precision-recall (PR) curve. We further use mean average precision (mAP) as defined in [31] and in Equation 10, 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.

$$AP = \frac{1}{n} \sum_{i=0}^{n} \frac{i}{r_i} \qquad (10)$$

Implementation details: For feature extraction, we use the Caffe framework [10], which provides pre-trained models on ImageNet for both employed CNN networks[1]. The implementation of the deep model is based on Theano [28]. For the three hidden layers [`fc_0`, `fc_1`, `fc_2`], we use [800, 400, 250] and [2500, 1000, 500] neurons for AlexNet and GoogleNet respectively. Thus, the dimensionality of the output embeddings is 250 and 500 dimensions for the two architectures respectively. Adam optimization [13] is employed with learning rate $l = 10^{-5}$ and mini-batches of size $m = 1000$ triplets. For the triplet generation, we set $t = 0.8$, which generates approximately 2k pairs in $\mathcal{P}$ and 7M and 5M triplets in $\mathcal{T}$, for AlexNet and GoogleNet, respectively. Other parameters are set to $\gamma = 1$ and $\lambda = 10^{-5}$.

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

4.2. Competing approaches

The proposed approach is compared against six approaches from the literature. Four of those were developed having access to the evaluation set. The remaining two do not require a development dataset. The first four approaches include the following:

Auto Color Correlograms (ACC): Cai *et al*. [1] use uniform sampling to extract one frame per second for the input video. The auto-color correlograms [8] 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*. [4] 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 programming (mPDP) and time-shift m-pattern similarity (TPS) to determine video similarity.

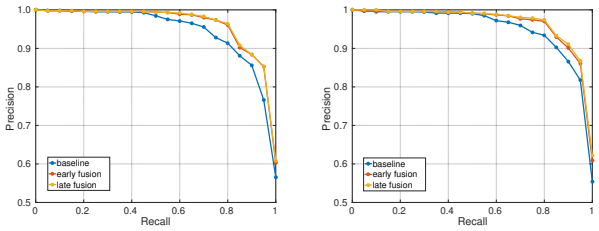
Layer-wise Convolutional Neural Networks (CNN-L): Kordopatis-Zilos *et al*. [14] extract the frame descriptors based on the same process as in Section 3.1 using GoogleNet. A video-level histogram representation derives from the aggregation of the layer vectors to visual words. The similarity between two videos is computed as the tf-idf weighted cosine similarity over the video-level histograms.

Stochastic Multi-view Hashing (SMVH): Hao *et al*. [7] 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 approaches are based on the work of Wu *et al*. [31]:

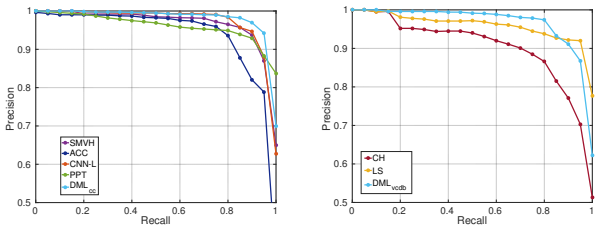
Color Histograms (CH): This is a global video representation based on the color histograms of keyframes. The color histogram is a concatenation of 18 bins for Hue, 3 bins for Saturation, and 3 bins for Value, resulting in a 24-dimensional vector representation for every keyframe. The global video signature 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 [12]).



(a) AlexNet (b) GoogleNet

Figure 5. Precision-Recall curve of the proposed approach based on the two CNN architectures and for the three system setups.



(a) Evaluation dataset (b) No/Other dataset

Figure 6. Precision-Recall curve of the proposed approach and state-of-the-art approaches, separated by the development dataset.

5. Experiments

5.1. Experimental results

In this section, we study the performance of the proposed approach in the CC_WEB_VIDEO dataset in relation to the underlying CNN architecture and the different fusion schemes. AlexNet and GoogleNet, two popular CNN architectures, are benchmarked. For each of them, three configurations are tested: i) **baseline**: fuse all frame descriptors to a single vector and use it for retrieval without any transformation, ii) **early fusion**: fuse all frame descriptors to a single vector and then apply the learned embedding function to generate the video descriptor for retrieval, iii) **late fusion**: apply the learned embedding function to every frame descriptor and fuse the embeddings to derive video representations for retrieval.

Figure 5 and Table 1 illustrate the PR curves and the mAP, respectively, of the two CNN architectures with the three system setups. Late fusion runs outperform both baseline and early fusion ones for both CNN architectures. GoogleNet achieves better results for all three settings with considerable margin, with precision more than 97% up to 80% recall and mAP scores of 0.968 and 0.969 for early and late fusion respectively. Both fusion schemes clearly improve the performance of the baseline approach for both architectures. Both schemes achieve very similar results, which indicates that the choice of the employed fusion scheme is not crucial for the performance of the method.

Architecture	baseline	early fusion	late fusion
AlexNet	0.948	0.964	0.964
GoogleNet	0.952	0.968	**0.969**

Table 1. mAP of both CNN architectures based on the baseline and two DML fusion schemes.

5.2. Comparison of different features

To delve deeper into performance, we study the performance of the DML framework with early fusion built on features extracted based on three different methods. The benchmarked methods are: i) **proposed**: apply max-pooling to all convolution layers and concatenate the vectors, ii) **last conv**: apply max-pooling to the activations of the last convolution layer, iii) **first fc**: the activations of the first fully-connected layer. We experiment with both CNN architectures.

Table 2 depicts the mAP of the three feature extraction methods for two CNN architectures. The proposed feature extraction scheme outperforms the runs of the compared feature extraction methods, for both architectures. In case of AlexNet, the **proposed** method marginally outperforms the **first fc** method. But, our approach reports clearly better performance compared to the others when GoogleNet is used. Hence, we may draw the conclusion that the feature extraction using all convolution layers yields better results for NDVR. Additionally, the triplet loss training scheme clearly improves performance compared to the baseline of section 5.1.

Architecture	proposed	last conv	first fc
AlexNet	0.964	0.957	0.962
GoogleNet	**0.968**	0.960	0.961

Table 2. mAP of three feature extraction methods for the two CNN architectures.

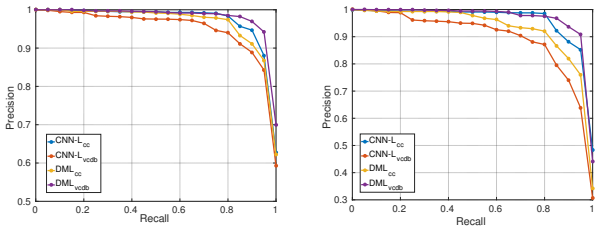
5.3. Comparison against NDVR state-of-the-art

For comparing the performance of our approach with the six NDVR approaches from the literature, we select the setup using GoogleNet features and late fusion denoted as DML$_{vcdb}$, since it achieved the best results. For the sake of comparison and completeness, we further provide the results of our model trained on a triplet set derived from both VCDB (similar to DML$_{vcdb}$) and also videos sampled from CC_WEB_VIDEO, denoted as DML$_{cc}$. The latter simulates the situation where the DML-based approach had access to a portion of the evaluation corpus, similar to the setting used by the competing approaches.

Table 3 presents the mAP scores of the competing methods. The methods are grouped based on the dataset used during development. Our approach outperforms all methods in each group with a clear margin. The same result derived from the comparison of the PR curves is illustrated in Figure 6, with the light blue line (proposed approach)

	Evaluation Dataset					No/Other Dataset		
Method	**ACC**	**PPT**	**SMVH**	**CNN-L**	**DML$_{cc}$**	**CH**	**LS**	**DML$_{vcdb}$**
mAP	0.944	0.958	0.971	0.974	**0.981**	0.892	0.954	**0.969**

Table 3. mAP comparison between two variants of the proposed approach against six state-of-the-art methods. The approaches are divided based on the dataset used for development.



(a) CC_WEB_VIDEO (b) CC_WEB_VIDEO*

Figure 7. Precision-Recall curve comparison of the proposed approach with two variants of [14] on two dataset setups.

lying upon all others up to 90% recall in both cases. It is noteworthy that our approach trained on VCDB dataset outperforms four out of six methods, with two approaches achieving marginally better results, but both developed on the evaluation dataset.

5.4. Performance in the presence of distractors

In our last experiment, we implemented the second best performing approach CNN-L [14] based on information derived from the VCDB dataset, i.e. we built the layer codebooks from a set of video frames sampled from the aforementioned dataset. We then tested two variations, the CNN-L$_{cc}$ that was developed on the CC_WEB_VIDEO dataset (same as Section 5.3) and the CNN-L$_{vcdb}$ 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 (the videos from the other subsets are considered to be dissimilar).

Figure 7 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 CNN-L approach, for both dataset setups. The proposed approach outperforms the CNN-L approach for all runs and setup at any recall point by a large margin. Similar conclusions can be drawn from the mAP scores of Table 4. The performance of CNN-L drops by more than 0.02 and 0.062 when it is trained on VCDB, for each setup respectively. Again, there is a considerable drop in performance in CC_WEB_VIDEO* setup for both approaches, with the proposed being more resilient to the setup change. As a result, the proposed approach has been demonstrated to be highly competitive and possible to transfer to different datasets with comparatively lower performance loss.

Run	CC_WEB_VIDEO	CC_WEB_VIDEO*
CNN-L$_{vcdb}$	0.954	0.898
DML$_{vcdb}$	**0.969**	**0.934**
CNN-L$_{cc}$	0.974	0.960
DML$_{cc}$	**0.981**	**0.970**

Table 4. mAP comparison of the proposed approach with two variants of the approach [14] on two different dataset setups.

6. Conclusions and Future Work

We presented a new video-level representation for Near-Duplicate Video Retrieval, which leverages the effectiveness of features extracted from intermediate convolution layers and Deep Metric Learning. We proposed a DML architecture based on video triplets and a novel triplet generation scheme that generates a compact video-level representation for the NDVR problem. The proposed approach was tested on two CNN architectures and exhibited highly competitive performance when developed on an independent dataset from the evaluation set. Furthermore, it outperformed all compared approaches from the literature by a clear margin. Finally, the performance of the proposed approach was compared with the best method from state-of-the-art in terms of Precision-Recall and mAP and in two different setups of CC_WEB_VIDEO dataset.

In the future, we plan to look into further improvements to the proposed approach, e.g. by considering more effective fusions schemes (compared to early and late fusion) and by training the DML architecture end-to-end (instead of using features from pre-trained CNN architectures). Moreover, we are going to conduct more comprehensive evaluations using more challenging datasets, and we will also assess the applicability of the approach on the problem of Partial Duplicate Video Retrieval (PDVR).

7. Acknowledgments

This work is supported by the InVID project, partially funded by the European Commission under contract numbers 687786.

References

[1] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X.-S. Hua, and S. Li. Million-scale near-duplicate video retrieval system. In *Proceedings of the 19th ACM international conference on multimedia*, pages 837–838. ACM, 2011.

[2] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

[3] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1565–1573, 2015.

[4] C.-L. Chou, H.-T. Chen, and S.-Y. Lee. Pattern-based near-duplicate video retrieval and localization on web-scale videos. *IEEE Transactions on Multimedia*, 17(3):382–395, 2015.

[5] M. Douze, H. Jégou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Transactions on Multimedia*, 12(4):257–266, 2010.

[6] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[7] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas. Stochastic multiview hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*, 19(1):1–14, 2017.

[8] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3):245–268, 1999.

[9] Z. Huang, H. T. Shen, J. Shao, X. Zhou, and B. Cui. Bounded coordinate system indexing for real-time video clip search. *ACM Transactions on Information Systems (TOIS)*, 27(3):17, 2009.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[11] Y.-G. Jiang, Y. Jiang, and J. Wang. Vcdb: a large-scale database for partial copy detection in videos. In *European Conference on Computer Vision*, pages 357–371. Springer, 2014.

[12] Y. Ke and R. Sukthankar. 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*, volume 2, pages II–II. IEEE.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *International Conference on Multimedia Modeling*, pages 251–263. Springer, 2017.

[15] W. Kraaij and G. Awad. Trecvid 2011 content-based copy detection: Task overview. *Online Proceedings of TRECVid*, 2011.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys (CSUR)*, 45(4):44, 2013.

[18] L. Liu, W. Lai, X.-S. Hua, and S.-Q. Yang. Video histogram: A novel video signature for efficient web video duplicate detection. In *International Conference on Multimedia Modeling*, pages 94–103. Springer, 2007.

[19] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012.

[20] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1855, 2015.

[21] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision*, pages 3–20. Springer, 2016.

[22] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574*, 2014.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[24] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua. Real-time large scale near-duplicate web video retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 531–540. ACM, 2010.

[25] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 423–432. ACM, 2011.

[26] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*, 15(8):1997–2008, 2013.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[28] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[29] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 1386–1393, 2014.

[30] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao. Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 153–162. ACM, 2013.

[31] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 218–227. ACM, 2007.

[32] L. Yang. Distance metric learning: A comprehensive survey. 2006.

[33] J. R. Zhang, J. Y. Ren, F. Chang, T. L. Wood, and J. R. Kender. Fast near-duplicate video retrieval via motion time series matching. In *2012 IEEE International Conference on Multimedia and Expo*, pages 842–847. IEEE, 2012.

[34] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian. Good practice in cnn feature transfer. *arXiv preprint arXiv:1604.00133*, 2016.

[35] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*, pages 649–656. IEEE, 2011.