Attention-Based Deep Metric Learning for Near-Duplicate Video Retrieval

Kuan-Hsun Wang*, Chia-Chun Cheng*, Yi-Ling Chen†, Yale Song†, Shang-Hong Lai*†
*National Tsing Hua University, Hsinchu, Taiwan, †Microsoft Corporation

***Abstract*—Near-duplicate video retrieval (NDVR) is an important and challenging problem due to the increasing amount of videos uploaded to the Internet. In this paper, we propose an attention-based deep metric learning method for NDVR. Our method is based on well-established principles: We leverage two-stream networks to combine RGB and optical flow features, and incorporate an attention module to effectively deal with distractor frames commonly observed in near duplicate videos. We further aggregate the features corresponding to multiple video segments to enhance the discriminative power. The whole system is trained using a deep metric learning objective with a Siamese architecture. Our experiments show that the attention module helps eliminate redundant and noisy frames, while focusing on visually relevant frames for solving NDVR. We evaluate our approach on recent large-scale NDVR datasets, CC_WEB_VIDEO, VCDB, FIVR and SVD. To demonstrate the generalization ability of our approach, we report results in both within- and cross-dataset settings, and show that the proposed method significantly outperforms state-of-the-art approaches.**

***Index Terms*—Near-duplicate video retrieval, video copy detection, deep metric learning, large-scale evaluation**

I. Introduction

Near-duplicate video retrieval (NDVR) has received great attention in recent years due to an increasing number of illegal, pirate videos uploaded to the Internet. These videos contain the content of original videos but make them look subtly different in order to sidestep automatic copy detection systems. This is commonly done by applying various tricks such as changing the aspect ratio, scale, color, frame rate, and applying padding and overlaying text. Some smart combinations of these can make it difficult to detect near-duplicate videos.

NDVR has been well-studied in the literature. In 2008, NIST launched the Content-based Copy Detection (CCD) challenge [1]. Several algorithms were proposed to compete in the challenge and near-perfect performance has been reported just under four years [2]. Therefore, NDVR was once treated a solved problem. However, recent large-scale real-world datasets, such as VCDB [3], FIVR [4] and SVD [5], have shown that traditional approaches to NDVR are ineffective in dealing with highly complex nature of the real-world copy patterns. This made the community reconsider NDVR as a challenging problem again.

In this work, we focus on addressing three key challenges in the real-world NDVR. First, traditional approaches [2], [6], [7] typically compare videos at the *frame-level* and require post-processing to aggregate per-frame results, *e.g.*, the temporal alignment network [8]. To capture the temporal context, we build our network based on a two-stream architecture that combines RGB and optical flow features. Second, real-world duplicate videos often contain frames that are *noisy – i.e.*, frames heavily edited from the original content – and *irrelevant – i.e.*, unrelated frames deliberately inserted to circumvent copy detection. To deal with this, we develop an attention-based video encoder that aggregates frame-level information while avoiding noisy and irrelevant frames. Lastly, traditional *"modularized"* systems that combine separately developed components, *e.g.*, [9], are ineffective in dealing with the complex patterns in real-world NVDR. To learn the real-world copy patterns, we design our architecture to be a Siamese network and train it with a triplet ranking loss based on a real-world database containing pairs of original and copy videos.

Our approach has several benefits over the traditional, frame-wise, and modularized methods [2], [6], [7], [9], [10]. Our network combines both RGB and optical flow information into our video representation. This makes our approach robust to "static" distractors – *e.g.*, black padding and text overlay – that are difficult to rule out only from RGB features. Also, our approach does not require post-processing and instead *learns how to aggregate* frame-level features via attention. We show that our approach achieves state-of-the-art results on challenging real-world NVDR datasets.

Our main contributions include: 1) We propose a two-stream approach to utilize RGB and optical flow features to learn discriminative video representation for NVDR; 2) We incorporate an attention module to aggregate the most relevant information for NVDR; and 3) we propose an effective feature aggregation approach to further enhance the discriminative power in our representation. We show that our method achieves the state-of-the-art result on VCDB [3], FIVR [4], and CC_WEB_VIDEO [11] datasets.

II. Related Work

NVDR methods typically consist of feature extraction and similarity search. Most methods focus on extracting discriminative features to improve the performance. Two types of features have been popular: local features and global features. Local features, such as SIFT [12] and local binary pattern (LBP) [13], preserve detailed information useful for detecting copy sequences but require a high computational cost and a large storage space. On the other hand, global features, such as color histogram, can tolerate some noisy frames but are sensitive to image shift and the padding of black margins.

A. Local feature-based methods

Douze *et al.* [2] extract SIFT and center-symmetric local binary pattern (CS-LBP) [14] features from every frame and match them with candidate copy segments. Jiang *et al.* [3] implement a system using SIFT descriptors and standard bag-of-words representation on every frame. In addition, they also use an inverted file structure to index SIFT features and exclude incorrect matches with geometric verification using weak geometric consistency (WGC). Finally, a temporal network finds the copied segment. Zhang *et al.* [15] use the fast CenSurE keypoint detector and BRIEF descriptors, and propose Binary Temporal Alignment to efficiently find a match between a reference and query videos. To compress the size of SIFT descriptors, Zhu *et al.* [16] propose the temporal-concentration SIFT (TC-SIFT) by tracking the SIFT features only at representative frames.

B. Global feature-based methods

Wu *et al.* [11] calculate a color histogram for every keyframe of a video as global signature. Esmaeili *et al.* [17] generate temporally informative representative images (TIRIs) from a video sequence, which contain spatial and temporal information. Uchida *et al.* [18] utilize an efficient DCT sign-based global feature to detect copy sequence. Guzman *et al.* [19] propose an approach based on multiple descriptors and use reinforcement learning to train the decision module.

Traditional NVDR methods aggregate frame-level features into video-level representation using Global Vector (GV) [11] and Bag-of-Words (BoW) [20]. GV averages frame-level features, treating every frame equally, while BoW maps every frame feature to a visual codebook and aggregates them as either a histogram or TF-IDF to obtain video-level representation. However, GV can easily be dominated by high-frequency patterns that appear across videos regardless of copy content, and thus have low discriminative power. Although BoW takes visual-word importance into account, e.g., by computing TF-IDF, it is sensitive to the dataset used to create the visual codebooks. In this work, we use an attention mechanism to aggregate frame-level features into video-level features.

C. CNN feature-based methods

Jiang *et al.* [21] propose two baselines using CNN features, standard CNN and Siamese CNN (SCNN). The standard CNN uses features extracted from an ImageNet pre-trained AlexNet [22], while SCNN uses copy/non-copy image patch pairs as training data and the contrastive loss as the objective function. Kordopatis *et al.* [4] use intermediate convolutional layers as feature descriptors. Additionally, two-layer aggregation techniques, vector aggregation and layer aggregation, are proposed to combine features from different layers. Wang *et al.* [23] obtain compact video representation from CNN features using principal component analysis (PCA) and sparse coding. Kordopatis *et al.* [24] use a video triplet as input to CNN and train it with a triplet loss to ensure that a positive-query distance is smaller than a negative-query distance.

In contrast to using a single feature, many algorithms utilizing multiple features are proposed in the recent years. Song *et al.* [6] argue that a single source of features is insufficient to represent video content. Therefore, Multiple Feature Hashing (MFH), which uses the local information (LBP) and global features (color histogram), are exploited for video representation.

In this paper, we use both RGB and optical flow to integrate spatial and temporal information. We incorporate an attention module to aggregate frame-level features into a global video representation while filtering out irrelevant information. We further enhance our video-level representation with multiple features induced by multi-head attention. In addition, we use online hard negative mining to encourage our network to learn from difficult examples.

III. Proposed Method

In this work, we aim to learn robust and discriminative video representation for real-world NVDR. Fig. 1 provides an overview of our network architecture. We uniformly sample frames from an input video and feed them into our feature extractors to compute RGB features and optical flow features. Real-world video copies often contain frames that are noisy and sometimes even irrelevant to the original video. To avoid these redundant features dominating the video feature, we incorporate an attention module to aggregate frame-level features into a global representation while avoiding noisy and irrelevant frames. In particular, we employ multi-head attention to derive multiple video-level representations, each attending to different parts of an input video. We train our network in a Siamese fashion to encourage the distance between a pair of duplicate videos to be closer than non-copy pairs.

A. Frame-Level Representation

To deal with videos of variable length, we uniformly sample M frames from a video and use Inception-V3 [26] pretrained on ImageNet [27] to extract frame-level RGB features. More specifically, the set of local frame-level features is formed by $\mathbf{X}_{RGB} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$, where $\mathbf{X}_{RGB} \in \mathbb{R}^{D_{RGB} \times M}$. We take the output of the last adaptive average pooling layer to obtain the RGB features, $D_{RGB} = 2048$. Inception-V3 is based on a varying size of convolutional kernels, and thus helps capture both low-level features (texture, shape) and high-level features (abstract concepts about objects), which is adequate for video copy detection.

Optical flow features have been shown to play an important role in video tasks, such as action recognition [28], [29]. In addition to RGB features, we use FlowNet 2.0 [30] pretrained on MPI-Sintel [31] to extract optical flow features $\mathbf{X}_{flow} \in \mathbb{R}^{D_{flow} \times M}$, $D_{flow} = 1024$. We combine $\mathbf{X}_{RGB}$ and $\mathbf{X}_{flow}$ by an attention module to form the final video representation, which we describe next.

B. Video-Level Representation

Real-world NVDR is challenging due to *noisy* (*e.g.*, heavily edited and/or affine transformed) and *irrelevant* (*e.g.*, deliberated inserted to circumvent detection) frames in copied videos.

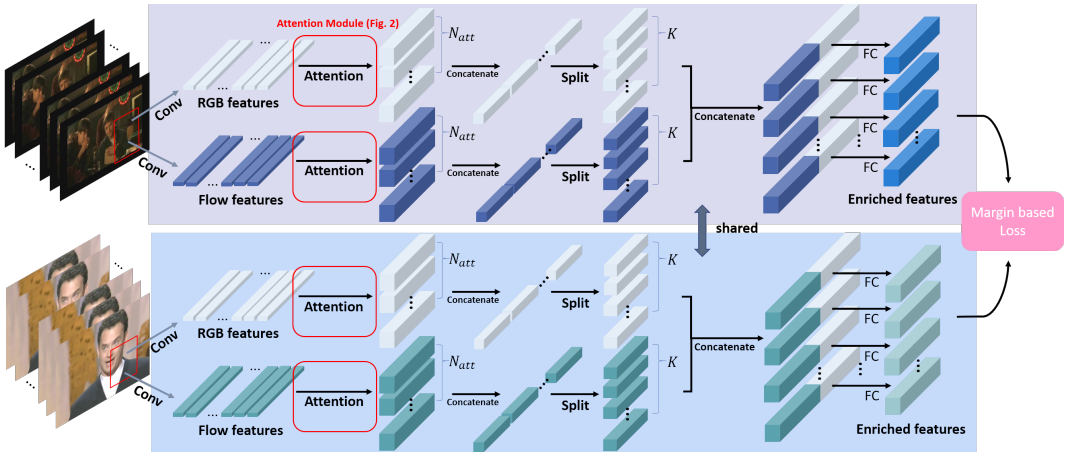

Fig. 1: We use a two-stream network to combine RGB and optical flow, and incorporate multi-head attention with shifting operations to aggregate features from the most relevant frames. We train our network in a Siamese fashion with a margin-based triplet ranking loss.

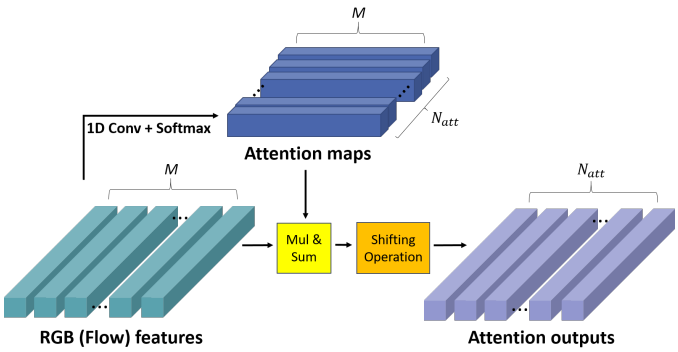

Fig. 2: Illustration of our attention module. It consists of N_{att} attention heads (defined as 1D conv + SoftMax) and outputs a $N_{att} \times M$ attention maps. We apply the shifting operation [25] to aggregate the $D \times M$ input features into $D \times N_{att}$ features.

Also, the variability in video lengths makes it difficult to detect copied segments: Video lengths vary from just a few seconds to several minutes/hours, and copied segments can appear only partially in long videos. Intuitively, not all frames are necessary to solve NVDR; it is rather beneficial to focus on a few most relevant frames/segments.

Neural attention mechanisms have been shown to be effective at learning representations that focus on the most relevant part of the input signal [25], [32]–[34]. Motivated by this, we incorporate an attention module to find the most relevant frame-level features and aggregate them to get robust video representation. Especially, we adopt multi-head attention clusters [25] to aggregate frame-level features and form a video-level *enriched feature set* (EFS).

Our attention module, shown in Fig. 2, is composed of N_{att} attention heads based on a 1D convolutional layer followed by a softmax layer (we set $N_{att} = 4$ in all our experiments). Each of the N_{att} attention heads sums up to 1.0 and determines the weights among the frame-level features in the video-level representation,

$$\mathbf{v} = \mathbf{X} \cdot \mathbf{a}, \tag{1}$$

where $\mathbf{a}$ is an M-dimensional attention vector. $\mathbf{X}$ can be either $\mathbf{X}_{RGB}$ or $\mathbf{X}_{flow}$ and thus the aggregated feature $\mathbf{v}$ is of dimension D_{RGB} or D_{flow} accordingly. To prevent the attention units from focusing on similar signals, we adopt the *shifting operations* [25] and modify Equation 1 as follows:

$$\mathbf{v} = \frac{\alpha \cdot \mathbf{Xa} + \beta}{\sqrt{N_{att}} \|\alpha \cdot \mathbf{Xa} + \beta\|_2}, \tag{2}$$

where α and β are learnable parameters and act as a linear transformation in the feature space. The transformed feature is first L_2 normalized and then undergoes another global L_2 normalization among all attention units by the factor of $\frac{1}{\sqrt{N_{att}}}$. Intuitively, the shifting operation enables different attention units to diverge from each other and the scale-invariance facilitates the optimization of the network.

To obtain the EFS, we concatenate N_{att} outputs of the attention module and partition it into $\mathcal{K}$ parts of equal length. The split feature vectors are then passed to $\mathcal{K}$ fully connected layers to obtain the final feature set for the NDVR task. Note that when $N_{att} < \mathcal{K}$, the individual features in EFS only attend to part of input video, while in the case of $N_{att} > \mathcal{K}$, the individual features further aggregate the outputs of the attention module to form more condensed representation. Under such setting, the EFS is comprised of multiple features that attend to different levels of granularity of the video-level representation according to the number of $\mathcal{K}$. We empirically show how different $\mathcal{K}$ affects the performance in our experiments.

C. Distance Metric

We adopt the cosine distance to measure the similarity between a query video $\mathcal{Q}$ and a reference video $\mathcal{R}$,

$$Dist\left(f_\theta(\mathcal{Q}), f_\theta(\mathcal{R})\right) = 1 - \frac{f_\theta(\mathcal{Q}) \cdot f_\theta(\mathcal{R})}{||f_\theta(\mathcal{Q})||_2 \cdot ||f_\theta(\mathcal{R})||_2} \tag{3}$$

where f_θ represents the embedding function. Because we are using multiple features from EFS that focus on different parts of videos, we compute the distance between videos differently during training and testing. During training, we aggregate the distances computed over multiple features as

$$Dist_{train}\left(\mathcal{Q},\mathcal{R}\right)=\sum_{i=1}^{\mathcal{K}}Dist\left(f_i(\mathcal{Q}),f_i(\mathcal{R})\right), \tag{4}$$

where f_i represents the i-th embedding feature, and $\mathcal{K}$ represents the size of the enriched feature set.

On the other hand, during testing, we find only the closest features to be the distance for video copy detection.

$$Dist_{test}\left(\mathcal{Q},\mathcal{R}\right)=\min_{\forall 0\leq i,j<\mathcal{K}}Dist\left(f_i(\mathcal{Q}),f_j(\mathcal{R})\right)). \tag{5}$$

We consider a query video $\mathcal{Q}$ a duplicate of $\mathcal{R}$ if the distance is smaller than a predefined threshold; we set it to 0.5.

D. Loss Function

To learn an embedding space in which near-duplicate videos have closer distance than dissimilar videos, we design our objective function to enforce the following constraint:

$$Dist\left(f_\theta(\mathcal{Q}),f_\theta(\mathcal{R}^+)\right)<Dist\left(f_\theta(\mathcal{Q}),f_\theta(\mathcal{R}^-)\right),$$
$$\forall\mathcal{Q},\mathcal{R}^+,\mathcal{R}^-\begin{cases}\mathcal{Q},\mathcal{R}^+ \text{ are NDV pair}\\ \mathcal{Q},\mathcal{R}^- \text{ are non-NDV pair}\end{cases} \tag{6}$$

We use the adaptive margin-based loss [35] to define our objective function. As shown in Wu *et al.* [35], the standard contrastive loss for metric learning defines a constant margin for all pairs of negative samples, which forces them to be embedded in the same space as the positive samples. On the other hand, the standard triplet ranking loss is computed over $\mathcal{O}(n^2)$ pairs or $\mathcal{O}(n^3)$ triplets of samples, which is computationally expensive, and most samples do not contribute to learning once the network starts to converge.

The adaptive margin-based loss combines the benefits of the contrastive loss and the triplet ranking loss: it allows the embedding space to be arbitrarily distorted similar to the ranking loss, yet it enjoys the computational efficiency of the contrastive loss. Our loss is defined as

$$L_{margin}(f_\theta(\mathcal{Q}),f_\theta(\mathcal{R}))=\max\{0,\gamma+y_{\mathcal{QR}}(D_{\mathcal{QR}}-\delta)\}. \tag{7}$$

where D is the cosine distance between video $\mathcal{Q}$ and $\mathcal{R}$ in our embedding space, γ controls the margin of separation, δ determines the boundary between NDV pair and non-NDV pair, and $y_{\mathcal{QR}}\in\{-1,1\}$ is a label indicating the relationship between $\mathcal{Q}$ and $\mathcal{R}$.

How should we select $\mathcal{R}^-$? Negative sampling is challenging for NVDR due to the imbalanced nature of the problem. The total number of negative samples (dissimilar video pairs) is far higher than that of positive samples (near duplicate videos). Moreover, most of the negative samples are easy examples and thus do not contribute much to learning. Here, we apply online hard negative mining [36] to sample negatives in every mini-batch. To do this, we compute the pairwise distance between every pair of samples in a mini-batch and select the ones with the smallest distance. This helps our network converge faster and learn discriminative features.

E. Implementation Details

To form the local feature set of an input video, we uniformly sample 500 frames to extract the color and optical flow features, except for the SVD dataset [5] where we set $M=100$ because it is mainly composed of short video clips. As for data augmentation, we apply dropout to the input frames by randomly selecting up to 80% of the input frames and setting the corresponding frame-level features to zero. This encourages the network to make predictions based on different parts of videos. We train our model using the Adam optimizer with a learning rate of $1\times e^{-4}$, and weight decay of $1\times e^{-4}$. The size of mini-batch is set to 256 and we train our network on one NVIDIA GTX 1080Ti GPU for 20,000 iterations. The network is trained end-to-end except for the pre-trained RGB/flow feature extractors, which we freeze their weights throughout training. Based on preliminary experiments, we fix $\gamma=2.0$ and $\delta=1.2$ of Equation 7 throughout the experiments.

IV. Experiments

We conduct experiments on three public datasets and compare our proposed approach with state-of-the-art methods.

a) ***VCDB [3]:*** This dataset has two parts: the "core" part contains 528 videos and 9,236 copy segment pairs, and the "distractor" part contains additional 100K videos that do not contain copied segments and merely serve as background distractor. All the videos are crawled using 28 query keywords.

b) ***FIVR-200K [37]:*** This dataset contains 225,960 videos collected from YouTube. The videos are categorized into three types: duplicate scene videos (DSV), complementary scene videos (CSV), and incident scene videos (ISV). We focus on the DSV subset that includes 7,257 videos.

c) ***CC_WEB_VIDEO [11]:*** This dataset contains 12,790 videos collected from YouTube, Google Video, and Yahoo Video. It contains 3,481 near-duplicate videos, categorized into *formatting* duplicates that contain the same content with different encoding format, frame rate, bit rate, and frame resolution, and *content* duplicates that contain photometric variations, editing, and pixel modifications.

d) ***SVD [5]:*** This dataset is divided into three subsets: The "query set" contains 1,206 videos. The "labeled set" contains 34,020 videos (10,211 positive and 26,927 negative labeled video pairs). The "probable negative unlabeled set" contains 526,787 negative videos without human annotation. All videos are collected from Douyin.

We follow the suggested evaluation protocol of each dataset. For VCDB, we measure the segment-level precision (SP), recall (SR), and F_1 score between SP and SR:

$$SP=\frac{|\text{correctly retrieved segments}|}{|\text{all retrieved segments}|}, \tag{8}$$

$$SR=\frac{|\text{correctly retrieved segments}|}{|\text{groundtruth copy}|}, \tag{9}$$

TABLE I: Cross-dataset results on VCDB. $\mathcal{K}$ is the size of the enriched feature set used in our method.

Method	Trained on	Precision	Recall	F1-score
Hough voting [3]	-	0.714	0.448	0.550
Temporal Network [3]	-	0.705	0.522	0.596
CNN [21]	VCDB	-	-	0.650
SCNN [21]	VCDB	-	-	0.631
CNN+SCNN [21]	VCDB	-	-	0.645
Compact CNN [23]	VCDB	-	-	0.704
LAMV [38]	YFCC100M	-	-	0.687
Ours (RGB only, $\mathcal{K}$=1)	FIVR-200K	0.728	**0.691**	0.709
Ours (RGB+Flow, $\mathcal{K}$=1)	FIVR-200K	0.867	0.610	0.717
Ours (RGB+Flow, $\mathcal{K}$=2)	FIVR-200K	**0.871**	0.631	0.732
Ours (RGB+Flow, $\mathcal{K}$=4)	FIVR-200K	0.794	0.685	0.735
Ours (RGB+Flow, $\mathcal{K}$=8)	FIVR-200K	0.847	0.674	**0.751**
Ours (RGB+Flow, $\mathcal{K}$=16)	FIVR-200K	0.838	0.660	0.738

$$F_1 = \frac{2 \cdot SP \cdot SR}{SP + SR} \tag{10}$$

For CC_WEB_VIDEO, FIVR-200K and SVD, we measure performance by mean average precision (mAP):

$$AP = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{r_i}, \tag{11}$$

where n is the number of near-duplicate videos to the query video and r_i is the rank of the i-th retrieved video.

A. Cross-dataset Evaluation

To demonstrate the generalization ability of our method to different datasets, we conduct cross-dataset evaluation, where we train a model on one dataset and test it on another.

a) ***VCDB [3]:*** We train our model on FIVR-200K and test it on VCDB. As for the baseline methods, Jiang *et al.* [3] introduce two approaches: Temporal Network [8] and Hough voting [10]; we compare with both approaches. The two approaches share the same pipeline. First, local SIFT features are extracted to encode video frames into BoW representation. The encoded frame signatures of all reference videos are then indexed in a database. Second, a query video is encoded into BoW representation to search for frame-level matching results in the database. Geometric verification is utilized to improve the matching results. Finally, temporal alignment methods, such as temporal network and Hough voting, are used to identify copy segments from frame-level matches.

We also compare with CNN-based approaches: CNN, Siamese CNN (S-CNN), and fusion of CNN and S-CNN all use deep features extracted from AlexNet [22] as frame-level features [21]. LAMV [38] learns a temporal layer to find temporal alignments. Compact CNN [23] takes advantage of CNN and sparse coding to obtain compact video representation.

Table I shows that our approach achieves competitive results in comparison with the state-of-the-art methods. Our method utilizes CNN as our frame-level features and incorporates an attention module, which helps our model focus on the most relevant frames and obtain more robust video features. We also show results with the different numbers of $\mathcal{K}$. Not surprisingly, using more embedding features generally improve the performance.

TABLE II: Cross-dataset results on FIVR-200K (mAP). BoW and LBoW are trained on FIVR-200K, while DML and ours are trained on VCDB; GV and HC do not require training. $\mathcal{K}$ is the size of the enriched feature set used in our method.

Features		Aggregation Method					
		GV	BoW	LBoW	DML	HC	Ours
Hand-crafted	HSV	0.167	0.202	N/A	0.163	0.360	-
	LBP	0.112	0.158		0.097		-
	ACC	0.196	0.240		0.182		-
	VLAD	0.294	0.323		0.285		-
CNN (RGB only, $\mathcal{K}$=1)	VGG	0.366	0.575	0.710	0.398	0.470	0.462
	INC	0.333	0.500	0.608	0.367		0.577
CNN (RGB+Flow, $\mathcal{K}$=1)	INC	-	-	-	-	-	0.596
CNN (RGB+Flow, $\mathcal{K}$=2)	INC	-	-	-	-	-	0.592
CNN (RGB+Flow, $\mathcal{K}$=4)	INC	-	-	-	-	-	0.619
CNN (RGB+Flow, $\mathcal{K}$=8)	INC	-	-	-	-	-	**0.627**
CNN (RGB+Flow, $\mathcal{K}$=16)	INC	-	-	-	-	-	0.616

b) ***FIVR-200K [37]:*** We train our model on VCDB and test it on FIVR-200K. We compare with five feature aggregation baselines to compute the similarity: Global Vectors (GV) [11], Bag-of-Words (BoW) [20], Layered Bag-of-Words (LBoW) [4], Deep Metric Learning (DML) [24] and Hashing Codes (HC) [6]. GV averages all frame-level features (either hand-crafted or CNN-based) into a single video representation. BoW uses visual word to compute the TF-IDF representation. DML is a supervised method and thus trained on VCDB in this experiment. HC projects every frame into the Hamming space and combine them into a video representation.

Table II shows mAP of all the baselines and our method. It shows that there is an obvious gap between traditional hand-crafted features and CNN features (both VGG [39] and Inception-V3 [26]). In addition, we observe further improvement in performance when jointly using RGB and optical flow.

The codebook of BoW and LBoW are trained on FIVR-200K, which accounts for their superior performance over other baseline methods. Although our model was trained solely on VCDB, we can achieve competitive results or outperform the BoW baseline. In addition, by increasing the size of the EFS ($\mathcal{K}$) to 8, our method achieves the best mAP. Generally, using more embedding features can store more information which should improve performance. However, more parameters need to be learned while training the model. Due to memory constraints, we need to reduce the batch size for training. As mentioned in Section III-E, we apply hard negative mining in each batch. Thus, at $\mathcal{K}$=16 we get a lower mAP due to a smaller batch size of 128.

c) ***CC_WEB_VIDEO [11]:*** Table III summarizes the mAP of all the baselines and our method. There are two evaluation settings, *intra*-query and *inter*-query. The main difference is that intra-query retrieves videos only from the corresponding subset of videos crawled using the same keyword. Inter-query, on the other hand, retrieves videos from the entire dataset, which is a more challenging setting; it is the same protocol used in Kordopatis *et al.* [24]. As shown in Table III, our method achieves competitive performance in both settings, outperforming all the baselines in the inter-query setting. This indicates that our method is robust in a more challenging scenario with a larger number of candidate videos.

TABLE III: Cross-dataset results (in mAP) on the CC_WEB_VIDEO. We report results under two settings: Intra-query retrieves videos from a subset of videos collected using the same keyword as the query video; intra-query retrieves videos from the entire dataset.

Method	Trained on	Intra-query	Inter-query
GV [11]	-	0.892	-
BoW [20]	-	0.944	-
LBoW [4]	-	0.954	0.898
HC [6]	-	0.958	-
DML [24]	VCDB	**0.969**	0.934
Ours (RGB only, $\mathcal{K}$=1)	VCDB	0.940	0.914
Ours (RGB+Flow, $\mathcal{K}$=8)	VCDB	0.964	**0.946**

TABLE IV: Within-dataset results on FIVR-200k.

Method	mAP
GV [11]	0.389
BoW [20]	0.302
LBoW [4]	0.362
DML [24]	0.465
HC [6]	0.468
Ours (RGB+Flow, $\mathcal{K}$=1)	0.527
Ours (RGB+Flow, $\mathcal{K}$=2)	0.532
Ours (RGB+Flow, $\mathcal{K}$=4)	0.550
Ours (RGB+Flow, $\mathcal{K}$=8)	0.583
Ours (RGB+Flow, $\mathcal{K}$=16)	**0.602**

B. Within-dataset Evaluation

a) ***FIVR-200K [37]:*** We split the FIVR-200K dataset into train and test splits, according to [37], and train/test our model on them. Following [37], we use ImageNet-pretrained VGG features [39] for all the baselines; ours use ImageNet-pretrained Inception-V3 features [26]. Table IV shows that our approach significantly outperforms the baseline methods by a large margin, including BoW and LBow. Compared with the best performing baseline, Hashing Codes [6], our method yields a significant improvement of 0.13 in mAP.

b) ***SVD [5]:*** Following [5], we select 1,000 query videos and their corresponding labeled videos as a training set. The remaining 206 query videos are utilized as the test set. During testing, the corresponding labeled videos of the 206 query videos and the probable negative unlabeled set are utilized as the database. As shown in Table V, when only the labeled set is included in the database, our method achieves the mAP of 91.4%. As more disctractor videos are added to the database, our system performance degrades but not drastically. We also include two top performing baselines reported in [5] in Table V. Although our method is not directly comparable to these baselines due to the different sizes of the database, our method still demonstrates competitive results considering the amount of unlabeled videos that are already added to the database.

C. Space and Time Complexity

For a practical video retrieval system, both accuracy and efficiency are important. We showed that our method achieves superior performance than baselines in terms of accuracy.

TABLE V: Evaluation results (mAP) on SVD dataset.

Method	Database Size	mAP
LBoW [4]	532,738	0.556
DML [24]	532,738	0.785
Ours (RGB+Flow, $\mathcal{K}$=8)	5,951	0.914
Ours (RGB+Flow, $\mathcal{K}$=8)	25,945	0.902
Ours (RGB+Flow, $\mathcal{K}$=8)	41,762	0.896
Ours (RGB+Flow, $\mathcal{K}$=8)	61,763	0.890
Ours (RGB+Flow, $\mathcal{K}$=8)	81,764	0.885

Here, we discuss the space and time complexities. The storage requirement grows linearly with $\mathcal{K}$, the number of embeddings we use in the final representation. Each embedding vector is of 4,096 dimension. Thus, we need $4 \times \mathcal{K} \times 4096$ bytes for each video. The time complexity is also proportional to $\mathcal{K}$. In our retrieval step, we compute pairwise distances between a query video and all the videos in the database. Thus, the time complexity is $\mathcal{O}(n\mathcal{K})$ for one query video.

D. Visualization

a) Attention: To better understand how the attention network has learned to assess the importance of features from different parts of a video, we do a side-by-side comparison of the keyframes from a query video and the top ranked retrieved videos.

Fig. 3 highlights the keyframes with high attention scores. We see that the attention network has successfully selected representative keyframes among the query and reference videos that are visually similar. Additionally, thanks to the shift operations in the attention network, the video-level representation can capture multiple representative frames across the video (not shown in the figure). This is beneficial since the video copy segment may stretch across multiple shots.

The bottom part of Fig. 3 demonstrates a challenging case of NDVR on real-world videos. Query B is a famous scene from the movie Titanic and there are several camcorded versions existing in the database. Despite that frames in the copied video undergo severe visual distortion (*e.g.*, the highlighted keyframes in B1 and B2), the attention network is still able to capture the transformation between duplicate video pairs.

b) Video Feature Space: Our approach learns an embedding space where near-duplicate videos stay close to each other. To validate this, we visualize our embedding space by projecting the learned features into 2-dimensional space using t-Distributed Stochastic Neighbor Embedding (t-SNE) [40]. Fig. 4 (a) shows the t-SNE 2D projection of VCDB core dataset using our network trained on FIVR-200K. We can see that the videos with same label (NDV) form clusters around each other. Fig. 4 (b) shows a similar visualization of the embedding space on FIVR-200K trained on VCDB. To reduce visual clutter, we plot only a subset of duplicate video labels and also reduce the number of videos belonging to each label. We observe similar clustering patterns in the model's embedding space.

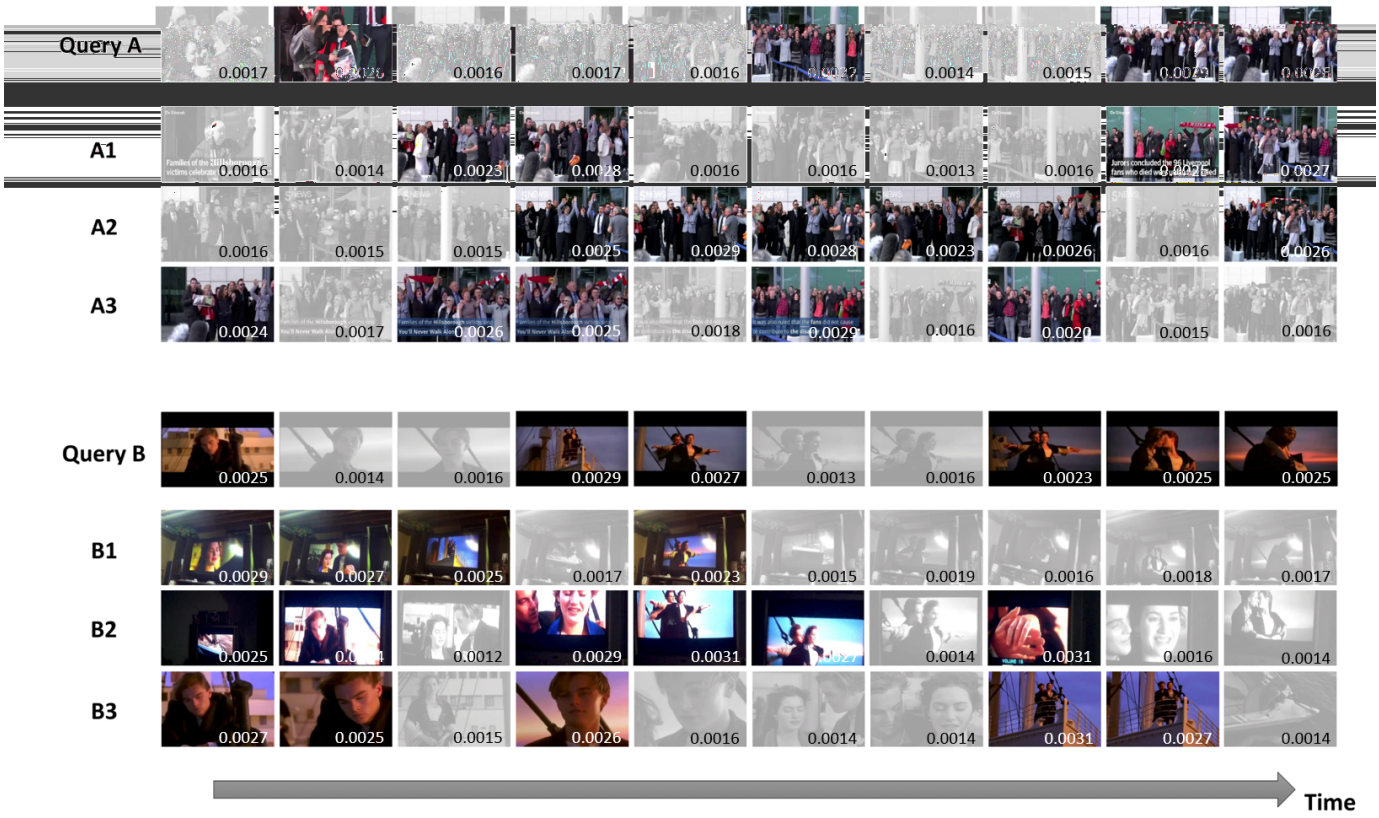

Fig. 3: Visualization of attention output. We show top-3 retrieved videos for each query video; each row shows 10 frames evenly spaced in time. We highlight frames that received the highest (top 1%) attention scores from each video.

V. Conclusion

We presented an attention based deep metric learning approach for near duplicate video retrieval (NVDR). We showed that our attention-based two-stream model learns video representation that is robust to complex transformation patterns in real-world NVDR. Moreover, we showed that representing a video with multiple features, computed from multi-head attention, allows our network to focus on only the most relevant segments in long videos, significantly improving performance on challenging real-world datasets. We evaluated our approach on four challenging real-world NVDR datasets – VCDB [3], FIVR-200K [37], CC_WEB_VIDEO [11], and SDV [5] – and achieved state-of-the-art results on most of the evaluated datasets.

References

[1] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in Proceedings of the 8th ACM international workshop on Multimedia information retrieval. ACM, 2006, pp. 321–330.

[2] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," IEEE Transactions on Multimedia, vol. 12, no. 4, pp. 257–266, 2010.

[3] Y.-G. Jiang, Y. Jiang, and J. Wang, "VCDB: a large-scale database for partial copy detection in videos," in ECCV. Springer, 2014, pp. 357–371.

[4] 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. Springer, 2017, pp. 251–263.

[5] Q.-Y. Jiang, Y. He, G. Li, J. Lin, L. Li, and W.-J. Li, "SVD: A large-scale short video dataset for near-duplicate video retrieval," in ICCV, 2019.

[6] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in ACM Multimedia, 2011, pp. 423–432.

[7] S. Wei, Y. Zhao, C. Zhu, S. Member, C. Xu, S. Member, and Z. Zhu, "Frame fusion for video copy detection," IEEE Trans. Circuits Syst. Video Technol, pp. 15–28, 2011.

[8] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua, "Scalable detection of partial near-duplicate videos by visual-temporal consistency," in ACM Multimedia, 2009, pp. 145–154.

[9] M. Hill, G. Hua, B. Huang, M. Merler, A. Natsev, J. R. Smith, L. Xie, H. Ouyang, and M. Zhou, "IBM Research TRECVID-2010 video copy detection and multimedia event detection system," 2011.

[10] M. Douze, H. Jégou, C. Schmid, and P. Pérez, "Compact video description for copy detection with precise temporal alignment," in ECCV, 2010, pp. 522–535.

[11] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in ACM Multimedia, 2007, pp. 218–227.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in ICCV, 1999.

[13] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," Pattern recognition, vol. 29, no. 1, 1996.

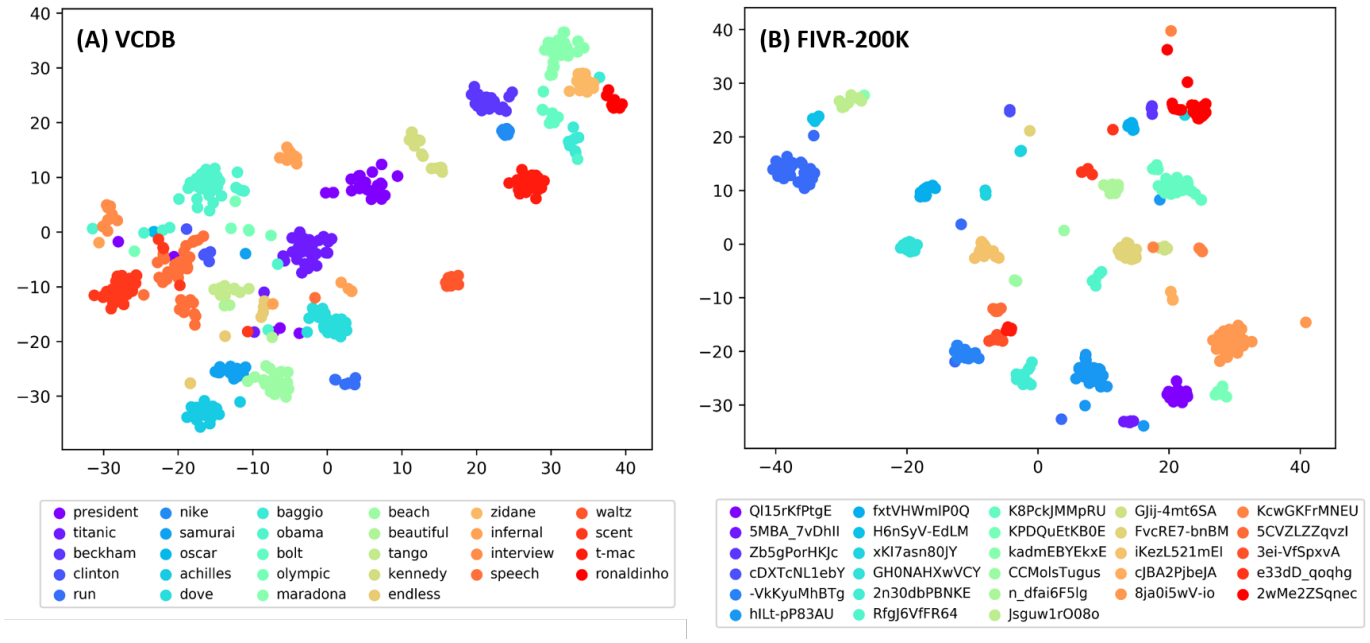

Fig. 4: tSNE visualization of video embedding space learned by our network on (a) VCDB and (b) FIVR-200K datasets. The embeddings are learned in the cross-dataset manner; VCDB is trained on FIVR-200K, and vice versa.

[14] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," Pattern Recognition, vol. 42, no. 3, pp. 425–436, 2009.

[15] Y. Zhang and X. Zhang, "Effective real-scenario video copy detection," in Pattern Recognition (ICPR). IEEE, 2016, pp. 3951–3956.

[16] Y. Zhu, X. Huang, Q. Huang, and Q. Tian, "Large-scale video copy retrieval with temporal-concentration sift," Neurocomputing, vol. 187, pp. 83–91, 2016.

[17] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," IEEE Transactions on information forensics and security, vol. 6, no. 1, pp. 213–226, 2011.

[18] Y. Uchida, K. Takagi, and S. Sakazawa, "Fast and accurate content-based video copy detection using bag-of-global visual features," in ICASSP, 2012, pp. 1029–1032.

[19] Z. J. Guzman-Zavaleta and C. Feregrino-Uribe, "Partial-copy detection of non-simulated videos using learning at decision level," Multimedia Tools and Applications, pp. 1–20, 2018.

[20] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X.-S. Hua, and S. Li, "Million-scale near-duplicate video retrieval system." in ACM Multimedia, 2011, pp. 837–838.

[21] Y.-G. Jiang and J. Wang, "Partial copy detection in videos: A benchmark and an evaluation of popular methods." IEEE Trans. Big Data, vol. 2, no. 1, pp. 32–42, 2016.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in NIPS, 2012, pp. 1097–1105.

[23] L. Wang, Y. Bao, H. Li, X. Fan, and Z. Luo, "Compact cnn based video representation for efficient video copy detection," in International Conference on Multimedia Modeling. Springer, 2017, pp. 576–587.

[24] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval with deep metric learning," in ICCVW, 2017, pp. 347–356.

[25] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in CVPR, 2018, pp. 7834–7843.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in CVPR, 2016, pp. 2818–2826.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in CVPR, 2009, pp. 248–255.

[28] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in NIPS, 2014, pp. 568–576.

[29] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in ECCV. Springer, 2016, pp. 20–36.

[30] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in CVPR, 2017, pp. 2462–2470.

[31] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in ECCV, 2012.

[32] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in ICCV, 2017.

[33] Y. Lou, Y. Bai, S. Wang, and L.-Y. Duan, "Multi-scale context attention network for image retrieval," in ACM Multimedia, 2018, pp. 1128–1136.

[34] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," arXiv, pp. arXiv–2001, 2020.

[35] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in ICCV, 2017, pp. 2840–2848.

[36] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in CVPR, 2015, pp. 815–823.

[37] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "FIVR: Fine-grained incident video retrieval," IEEE Transactions on Multimedia, pp. 1–1, 2019.

[38] L. Baraldi, M. Douze, R. Cucchiara, and H. Jégou, "LAMV: Learning to align and match videos with kernelized temporal layers," in CVPR, 2018, pp. 7804–7813.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ICLR, 2015.

[40] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.