VVS: Video-to-Video Retrieval with Irrelevant Frame Suppression

Won Jo[1], Geuntaek Lim[1], Gwangjin Lee[1], Hyunwoo Kim[1], Byungsoo Ko[2], Yukyung Choi[1]

[1]Sejong University
[2]NAVER Vision
{jwon, gtlim, gjlee, hwkim}@rcv.sejong.ac.kr, kobiso62@gmail.com, ykchoi@rcv.sejong.ac.kr

Abstract

In content-based video retrieval (CBVR), dealing with large-scale collections, efficiency is as important as accuracy; thus, several video-level feature-based studies have actively been conducted. Nevertheless, owing to the severe difficulty of embedding a lengthy and untrimmed video into a single feature, these studies have been insufficient for accurate retrieval compared to frame-level feature-based studies. In this paper, we show that appropriate suppression of irrelevant frames can provide insight into the current obstacles of the video-level approaches. Furthermore, we propose a Video-to-Video Suppression network (VVS) as a solution. VVS is an end-to-end framework that consists of an easy distractor elimination stage to identify which frames to remove and a suppression weight generation stage to determine the extent to suppress the remaining frames. This structure is intended to effectively describe an untrimmed video with varying content and meaningless information. Its efficacy is proved via extensive experiments, and we show that our approach is not only state-of-the-art in video-level approaches but also has a fast inference time despite possessing retrieval capabilities close to those of frame-level approaches. Code is available at https://github.com/sejong-rcv/VVS

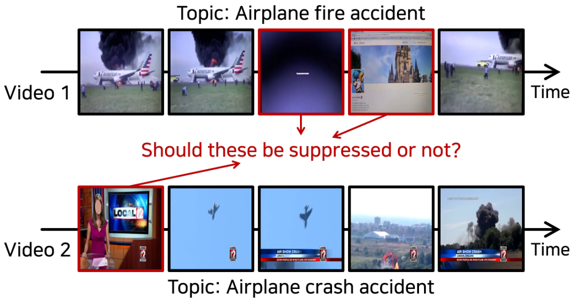
Topic: Airplane fire accident
Video 1
Time
Should these be suppressed or not?
Video 2
Time
Topic: Airplane crash accident

Figure 1: Q: Should the red boxes be suppressed? The red boxes in both videos should be excluded because they are unrelated to the topic in the video, although they are included for a specific purpose or reason. In this work, we demonstrate that the suppression of these red boxes enhances the distinctiveness of features when describing the entire video at once.

Introduction

Information retrieval is defined as finding the most relevant information in a large collection. It has evolved from finding text within a document (Griffiths, Luckhurst, and Willett 1986; Strzalkowski 1995; Bellot and El-Bèze 1999; Liu and Croft 2004) to finding images within an image set (Arandjelovic et al. 2016; Tolias, Sicre, and Jégou 2016; Jun et al. 2019; Ko et al. 2019; Ko and Gu 2020; Gu and Ko 2020). In recent years, with the fast growing trend of the video streaming market, several studies (Kordopatis-Zilos et al. 2017b, 2019b; Shao et al. 2021; Jo et al. 2022; Ng, Lim, and Lee 2022) have actively been conducted in content-based video retrieval (CBVR) to find desired videos from a set of videos.

The core of CBVR technology is to measure similarities between videos of different lengths, including untrimmed videos. This is divided into two streams according to the basic unit for measuring the similarity between two videos: a frame-level feature-based approach and a video-level feature-based approach. The former aggregates similarities between frame-level features in two videos to calculate a video-to-video similarity. Conversely, the latter describes each video as a single feature and computes a video-to-video similarity based on it. These two streams are in a trade-off relationship because the key foundation for determining similarity differs. The frame-level approach compares each frame directly; it is less dependent on factors such as video duration and whether or not it is trimmed. As a result, relatively accurate searches are possible, but processing speed and memory are expensive due to the necessity of numerous similarity computations and a considerable amount of feature storage space. In comparison, the video-level approach requires only one similarity calculation between a single pair of features, which is more efficient in terms of processing speed and memory. However, it is difficult to compress many frames of a video into a single feature, making approaches of this type generally inaccurate and sensitive to factors such as duration and trimness.

Ideally, if a video-level approach could be as distinct as a frame-level approach, it may be the best option in real-world scenarios. However, there are some problems that must be considered. First is that distractors in a video interfere with

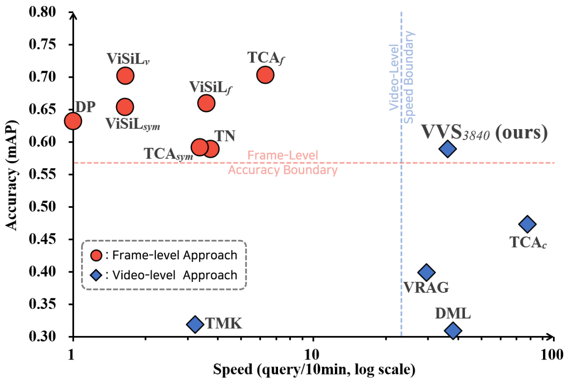

Figure 2: Speed-Accuracy Comparison on FIVR-200K. This is a comparison between the proposed approach and existing state-of-the-art approaches in terms of speed and accuracy on the FIVR-200K. Speed is represented by the average number of queries processed in 10 minutes, and accuracy is represented by the mAP in ISVR, the most difficult task.

the description of video-level features. Distractors in this context refer to frames with visual content that is unrelated to the main topic. Indeed, as shown in the two video examples in Figure 1, it is obvious that the red box frames corresponding to the distractors are not helpful for recognizing the topic of each video. We also present an experiment that demonstrates quantitative performance improvements when the distractors are manually eliminated from the previous video-level feature-based schemes in the supplementary material[1]. On the basis of these observations, this study proves that the description of video-level features with optimal suppression of distractors can be an ideal scenario for accurate and fast retrieval.

The objective of this work is to understand the significance of frames to determine how much they should be suppressed to produce a distinct video-level feature. To this end, we propose a Video-to-Video Suppression network (VVS). The VVS is an end-to-end framework consisting of two stages: an easy distractor elimination stage for removing frames that can be clearly recognized as distractors, and a suppression weight generation stage for determining how much to suppress the remaining frames via temporal saliency information and relevance of the topic. Our solution is the first explicitly designed framework that employs various signals for relevance, as opposed to earlier approaches (Kordopatis-Zilos et al. 2017b; Shao et al. 2021; Ng, Lim, and Lee 2022) where the model was implicitly intended to generate weights. As shown in Figure 2, VVS achieves state-of-the-art performance among video-level approaches, with search accuracy comparable to frame-level state-of-the-art performance while retaining competitive inference speed. In addition, extensive experiments included in the later section demonstrate the effectiveness of the proposed framework and the validity of the designed structure.

In summary, our main contribution is as follows: 1) we demonstrate that video-level features can be both accurate and fast with proper suppression of irrelevant frames, 2) we propose VVS, an end-to-end framework for embedding an untrimmed video as a video-level feature while suppressing frames via various signals, and 3) we show extensive experiments that demonstrate the effectiveness of our design, which acquires state-of-the-art performance.

Related Work

Frame-level Feature-based Approaches

There have been several recent studies in frame-level feature-based approaches. Dynamic Programming (DP) (Chou, Chen, and Lee 2015) detects a near-duplicate region by extracting the diagonal pattern from a frame-level similarity map. Temporal Network (TN) (Tan et al. 2009) distinguishes the longest route in a graph created by keypoint frame matching to discover visually similar frames between two videos, and Circulant Temporal Encoding (CTE) (Douze, M., Revaud, J., Verbeek, J., Jégou, H., & Schmid, C 2016) compares frame-level features using a Fourier transform. This allows frame information to be encoded in the frequency domain. The Video Similarity Learning (ViSiL) (Kordopatis-Zilos et al. 2019b) approach leverages metric learning by basing its operations on a frame-by-frame similarity map, while Temporal Nested Invariance Pooling (Jo et al. 2022) uses a local context-invariant property to design temporally robust pooling based on the standard (JTC1/SC29/WG11/N15339 2015). These approaches have higher accuracy than existing video-level approaches, but they are significantly slower in terms of search speed.

Video-level Feature-based Approaches

Various video-level approaches have also been explored in recent studies. Hashing Code (HC) (Song et al. 2013) collects and hashes a large number of local and global features to handle accuracy and scalability issues. Deep Metric Learning (DML) (Kordopatis-Zilos et al. 2017b) utilizes frame-level features from a layer codebook generated for intermediate Maximum Activation of Convolution (iMAC) (Kordopatis-Zilos et al. 2017a) features and fuses them to represent a video-level feature. Temporal Matching Kernel (TMK) (Poullot et al. 2015) generates a fixed length sequence for each video, regardless of the total number of frames in the video, using periodic kernels that take into account frame descriptors and timestamps. Furthermore, Learning to Align and Match Videos (LAMV) (Baraldi et al. 2018) designs a learnable feature transform coefficient based on TMK. Temporal Context Aggregation (TCA) (Shao et al. 2021) learns frame-level features into video-level features through self-attention and a queue-based training mechanism, while Distill-and-Select (DNS) (Kordopatis-Zilos et al. 2022) distills the knowledge of the teacher network, which is optimized from the labeled data, into a fine or coarse-grained student network to take further advantage of learning from the unlabeled data. This approach also maintains efficiency between the two types of students via a selector network. Video Region Attention

[1] Supplementary material can be found in the arxiv version: https://arxiv.org/abs/2303.08906

Graph (VRAG) (Ng, Lim, and Lee 2022) learns an embedding for a video by capturing the relationship of region units in frames via graph attention (Veličković et al. 2017) layers.

In general, these approaches can respond to a given query more quickly than frame-level approaches, even if the response is relatively inaccurate. However, our solution can respond as precisely as frame-level approaches while maintaining sufficient speed as a video-level approach. In addition, whereas DML, TCA, and VRAG (the most similar approaches to ours) ask FC layers, self-attention layers, and graph attention layers, respectively, to implicitly fuse frame-level features into a video-level feature (that is, only contrastive loss of fused features is used as the objective function), our approach is the first to generate video-level features via explicit signals, such as low-level characteristics, temporal saliency, and rough topic.

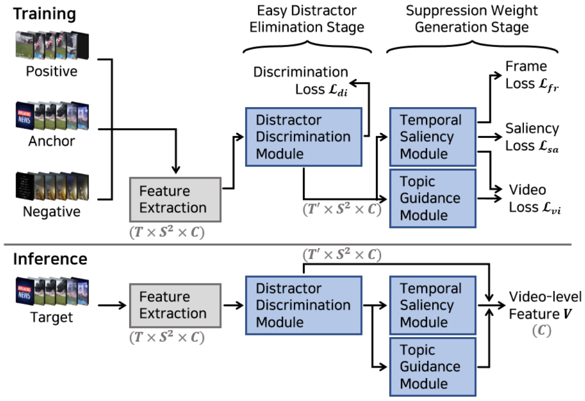
Training
Positive
Anchor
Negative
Easy Distractor Elimination Stage
Suppression Weight Generation Stage
Discrimination Loss $\mathcal{L}_{di}$
Frame Loss $\mathcal{L}_{fr}$
Saliency Loss $\mathcal{L}_{sa}$
Video Loss $\mathcal{L}_{vi}$
Feature Extraction
$(T \times S^2 \times C)$
Distractor Discrimination Module
$(T' \times S^2 \times C)$
Temporal Saliency Module
Topic Guidance Module
Inference
Target
Feature Extraction
$(T \times S^2 \times C)$
$(T' \times S^2 \times C)$
Distractor Discrimination Module
Temporal Saliency Module
Topic Guidance Module
Video-level Feature V
(C)

Figure 3: Pipeline Overview of VVS. The gray italic letters represent the size of the feature in each process.

Approach

Problem Formulation

Given a video with a duration of T, our goal is to embed it as a video-level feature V while suppressing frames corresponding to distractors. To determine which frame and to what extent it should be suppressed, video frames are first embedded in the frame-level features $X = \{x^{(t)}\}_{t=1}^{T}$ instead of being embedded directly in the video-level feature V. Next, T' frames are chosen by removing easy distractors that are readily identifiable as distractors due to a lack of information via the easy distractor elimination stage. In the subsequent suppression weight generation stage, weights $W = \{w^{(t)}\}_{t=1}^{T'}$ indicating the necessary degree of the remaining frames are calculated. Consequently, these weights are used to aggregate frame-level features into a video-level feature $V = \Psi(\{w^{(t)} \otimes x^{(t)}\}_{t=1}^{T'})$, where Ψ represents the Spatio-Temporal Global Average Pooling (ST-GAP) and $\otimes$ represents the Hadamard product. Figure 3 illustrates an overview of the VVS pipeline.

Feature Extraction

L_N-iMAC as a frame-level feature is first extracted for fair comparisons with many other works (Kordopatis-Zilos et al. 2017b, 2019b; Shao et al. 2021; Ng, Lim, and Lee 2022). Specifically, each frame is fed to the backbone network Φ as input, and Regional Maximum Activation of Convolution (R-MAC) (Tolias, Sicre, and Jégou 2016) is applied to its intermediate feature maps $\mathcal{M}^{(k)} \in \mathbb{R}^{S^2 \times C^{(k)} (k=1,\cdots,K)}$. Specifically, after obtaining feature maps from a total of K layers in Φ, N types of region kernels are used, depending on the granularity level, for applying R-MAC. As a result, each of the K intermediate feature maps $\mathcal{M}^{(k)}$ have their own channel $C^{(k)}$ but the same spatial resolution S^2. After these $\mathcal{M}^{(k)}$ are concatenated on the channel axis, they are generated as a frame-level feature $x \in \mathbb{R}^{S^2 \times C \, (C=\sum_{k=1}^{K} C^{(k)})}$. After applying Principal Component Analysis (PCA) whitening (Jégou and Chum 2012) to each of the features x, the L_N-iMAC feature $X \in \mathbb{R}^{T \times S^2 \times C}$ is obtained. Although the dimension of the channel axis could be reduced to different sizes for comparison with other approaches when applying PCA whitening, for convenience, the dimension of the frame-level feature is called C.

Easy Distractor Elimination Stage

In this section, we introduce the Distractor Discrimination Module (DDM), which eliminates frames that are clearly recognizable as distractors due to a lack of visual information. An easy distractor is a frame with little variation in pixel intensity and few low-level characteristics (edges, corners, etc.) in an image, such as the third frame of the first video in Figure 1. In the training phase of DDM, frame-level features corresponding to the easy distractor are injected into an input with a length of T, and the model is optimized to distinguish them. In the inference phase of DDM, frames predicted as easy distractors are removed from the input. This process results in the output length being longer than the input length T in the training phase but shorter in the inference phase. For convenience, the output length of DDM is always called T'. The overall flow is depicted in Figure 4.

Distractor Discrimination Module

To enable this module to learn to recognize an easy distractor, pseudo-labels are created using the magnitude of the frame-level features. This is because frames with few low-level characteristics have fewer elements to be activated from the backbone network of L_N-iMAC, which consists of several activation layers, resulting in a smaller magnitude of their intermediate feature map.

Specifically, before the training phase, a set of easy distractors with a magnitude lower than or equal to a magnitude threshold λ_{mag} is constructed from all frame-level features of the videos in the training dataset. Examples of easy distractors included in this set can be found in the supplementary material. During the training phase, features of easy distractors are picked from the set and randomly placed between the features X. In this case, only about 20–50% of T are injected, resulting in features of length T'. Simultaneously, the points where the distractors are injected are set at 0 and the opposite position at 1, resulting in a pseudo-label $Y_{di} = \{y_{di}^{(t)}\}_{t=1}^{T'}$. The injected features are

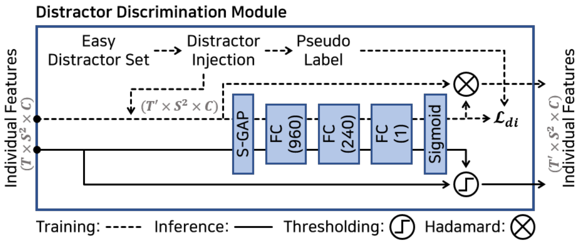

Figure 4: Pipeline of DDM. The gray italic letters indicate the size of the feature in each process. The number in parentheses in the layer blocks indicates the output dimension.

projected through multiple layers to calculate a confidence $W_{di} = \{w_{di}^{(t)}\}_{t=1}^{T'}$. Because only the components within each frame determine the criterion for identifying easy distractors, the multiple layers consist of only the Spatial Global Average Pooling (S-GAP) and FC layers to handle each frame independently without interaction between frames. As a result, this module is optimized by discrimination loss $\mathcal{L}_{di}$, computed as the binary cross entropy loss between the confidence W_{di} and the pseudo-label Y_{di}.

The objective of DDM is to convey features to the subsequent stage, erasing features of frames that are deemed to be easy distractors through thresholding for confidence. In this case, since the threshold operation is not differentiable during the training phase, the output is derived from the Hadamard product of the confidence W_{di} and the input features X, and during the inference phase, from a thresholding operation based on a distractor threshold λ_{di}.

Suppression Weight Generation Stage

Even if easy distractors are excluded through the previous stage, untrimmed videos still contain hard distractors that cannot be easily distinguished and are unrelated to the overall topic of the video due to the various content entanglements. In this section, the Temporal Saliency Module (TSM) and Topic Guidance Module (TGM) are introduced for calculating suppression weights, which indicate how close the remaining frames are to the hard distractor. TSM assesses the significance of each frame based on saliency information derived from frame-level similarities, while TGM measures the degree to which each frame relates to the overall topic of the video. The weights obtained from these two modules are converted into the suppression weights W using the Hadamard product.

Temporal Saliency Module

To measure the importance of each frame, saliency information is extracted in the training phase. This is inspired by ViSiL (Kordopatis-Zilos et al. 2019b), a model that refines a frame-level similarity map during training and accumulates it to a frame-level similarity via the Chamfer Similarity (CS) (Barrow et al. 1977) operation. Specifically, as the model is optimized, the CS operation leads to an increase in locations, which helps improve video-level similarity within a similarity map of a positive pair. Because of this, the increased locations contain the frames with a strong correlation between the positive pair (as proven in the supplementary material). Therefore, we propose a modified structure that can exploit this correlation as saliency information in TSM by extracting pseudo-labels based on these locations.

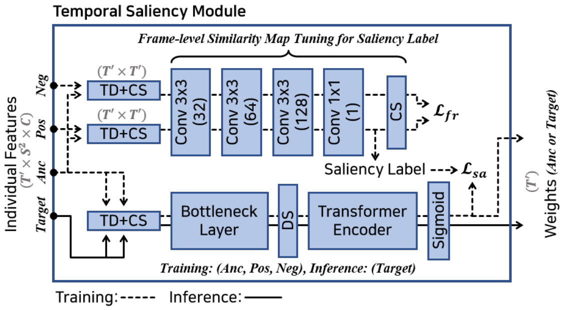

Figure 5: Pipeline of TSM. The gray italic letters represent the size of the feature in each process. The number in parentheses in the layer blocks indicates the output dimension.

Technically, as shown in Figure 5, frame-level features of the triplet are transformed by Tensor Dot (TD) and CS into a similarity map for the positive pair (i.e., anchor and positive) and a similarity map for the negative pair (i.e., anchor and negative). These similarity maps are then converted into tuned similarity maps $\mathcal{D}_p$ and $\mathcal{D}_n$ for the positive pair and the negative pair, respectively, through four convolutional layers. Here, we generate a pseudo-label Y_{sa} (i.e., saliency label) based on the increasing value within $\mathcal{D}_p$ in order to extract saliency information. This is formulated in Equation (1), where the superscript $\mathbf{T}$ is the transpose operation, and H is the Heaviside step function. Furthermore, ρ is the highest similarity of each frame in the anchor video for the positive video. The saliency label consists of values where ρ_i is 1 if it is greater than the average of ρ and 0 if it is less, thereby labeling the frame locations that indicate a strong correlation between the positive pair.

$$\begin{aligned} \rho_i &= \max_{j \in [1, T'']} \mathcal{D}_p^{(i,j)}, \\ \rho &= [\, \rho_1,\ \rho_2,\ \cdots,\ \rho_i,\ \cdots,\ \rho_{T''}]^{\mathbf{T}}, \\ Y_{sa} &= H(\rho - \frac{1}{T''}\sum_{i=1}^{T''} \rho_i) \in \mathbb{R}^{T''}. \end{aligned} \tag{1}$$

After completing the procedure for creating the saliency label, a self-similarity map is generated by applying TD and CS to two inputs consisting solely of the anchor. The self-similarity map is subsequently fed into the bottleneck layer, the transformer encoder, and the sigmoid to yield saliency weights $W_{sa} = \{w_{sa}^{(t)}\}_{t=1}^{T'}$, as shown in Figure 5. Here, only diagonal components are sampled from the output map of the previous layer to match the input format when entered into the transformer encoder, i.e., Diagonal Sampling (DS). Consequently, to enable TSM to recognize salient frames through training, the saliency loss $\mathcal{L}_{sa}$ is computed as the binary cross entropy loss between the saliency weights W_{sa} and the saliency label Y_{sa}, where the nearest interpolation is applied to the label to match the length of the output, T'.

The saliency loss $\mathcal{L}_{sa}$ is optimized with the frame loss $\mathcal{L}_{fr}$ for tuning the similarity map of the positive pair, which is covered in detail in the supplementary material. During the inference phase, only a self-similarity map for a given target video is fed into the layers to yield saliency weights.

Topic Guidance Module

The topic of the video is also one of the factors that determines the importance of frames. For this reason, we create an initial state I that gives direct, video-specific instruction on the topic to help the model generate guidance weights $W_{gu} = \{w_{gu}^{(t)}\}_{t=1}^{T'}$. More specifically, a rough topic representation G is initially constructed to roughly represent the topic of the video. According to the claim (Lin et al. 2017) that statistical moments (i.e., mean, max, etc.) have been mathematically proven to be invariant across multiple transformations, the ST-GAP, which consists of average operations, is used to create a $G \in \mathbb{R}^C$ that is robust to specific transformations between the frame-level features X. In fact, the topic of a video (even if untrimmed) is determined by what most of the content in that video represents. Therefore, since the average operation yields the direction in which most of the content vectors (i.e., frame-level features) point, an approximate (even if simple) representation of the topic can be obtained. As a result, the cosine similarity between G and X is employed to build the initial state $I \in \mathbb{R}^{T'}$, which guides the model to reference the topic. At this time, for convenience of operation, the S-GAP is applied to the frame-level features X to remove its spatial axis.

The initial state I is effective in directing the model in a rough pattern along the optimal path to the goal; however, a process of refinement must be added with the purpose of providing the guidance weights that more precisely suggest topic relevance. Thus, as illustrated in Figure 6, architecture is designed to refine the coarse pattern. With the initial state I of length T' as input, the data is collected by sliding 1×3 kernels in three convolutional layers, and then a 1×1 convolutional layer reduces the channel dimension. As the preceding three layers are traversed, the receptive field expands, indicating that the temporal spans of data gathered by these layers extend from the short-term to the long-term. Therefore, the output of the preceding three layers and the output of the 1×1 convolutional layer is designed to channel-wise concatenate, which is referred to as a hierarchical connection, to assist the model in grasping the topic relevance of each frame through direct utilization of the knowledge over various temporal spans. Then, a convolutional layer is applied to shrink the dimension of the channel axis. Only this module employs the tempered sigmoid proposed by (Papernot et al. 2021) rather than the sigmoid to reliably learn the weights from noises that may arise during the refining operation from rough patterns.

Video Embedding & Training Strategy

In the training phase, frame-level features are aggregated into a video-level feature $V \in \mathbb{R}^C$ by the Hadamard product with the suppression weights W calculated for each video in a triplet: an anchor, a positive, and a negative. At this time, in the case of positive and negative, only W_{gu} is used as the suppression weights W because their weights are not handled in TSM. As a result, the video loss $\mathcal{L}_{vi}$ is computed as the triplet margin loss between the three video-level features in the triplet. This loss, along with the three losses discussed above, optimize the model according to Equation (2) as,

$$\mathcal{L} = \mathcal{L}_{vi} + \mathcal{L}_{fr} + \mathcal{L}_{sa} + \alpha\mathcal{L}_{di}. \tag{2}$$

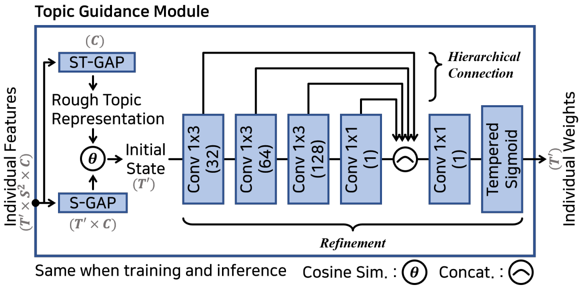

Figure 6: Pipeline of TGM. The gray italic letters represent the size of the feature in each process. The number in parentheses in layer blocks indicates the output dimension.

In addition, our approach follows the mining scheme of (Kordopatis-Zilos et al. 2019b) for videos consisting of triplets. α is a parameter for adjusting the learning of DDM as it is faster than other modules when observed empirically. Due to space limitations, further details can be found in the supplementary material.

Experiments

Evaluation Setup

Our experiments were evaluated on two retrieval settings[2] that are now widely used in CBVR: fine-grained incident video retrieval (FIVR) and near-duplicate video retrieval (NDVR). All performance evaluations are reported based on the mean average precision (mAP) (Zhu 2004), and the implementation details are covered in the supplementary material. Furthermore, VCDB (Jiang, Jiang, and Wang 2014) was used as a training dataset, and FIVR (Kordopatis-Zilos et al. 2019a) and CC_WEB_VIDEO (Wu et al. 2009) were used as evaluation datasets.

VCDB is aimed at video copy detection and consists of 528 core datasets with 9,236 partially copied pairs and about 100,000 videos with no additional metadata.

FIVR is equivalent to the FIVR task, which seeks videos connected to certain disasters, occurrences, and incidents. Furthermore, depending on the level of relevance desired, it is evaluated using three criteria: duplicate scene video retrieval (DSVR), complementary scene video retrieval (CSVR), and incident scene video retrieval (ISVR). In this dataset, there are two types in the family: FIVR-5K and FIVR-200K. FIVR-5K has 50 queries and 5,000 videos

[2] Some videos from EVVE (Revaud et al. 2013), a dataset for event video retrieval (EVR), another common evaluation setting, could not be downloaded. However, for further comparison, the benchmark for a subset we own (≈70.5% of the original) is covered in the supplementary material.

	Approach	*Dim.*	FIVR-200K		
			DSVR	CSVR	ISVR
frame	TN	-	0.724	0.699	0.589
	DP	-	0.775	0.740	0.632
	TCA_{sym}	1,024	0.728	0.698	0.592
	TCA_{f}	1,024	0.877	0.830	**0.703**
	TNIP	1,040	**0.896**	0.833	0.674
	ViSiL_{sym}	3,840	0.833	0.792	0.654
	ViSiL_{f}	3,840	0.843	0.797	0.660
	ViSiL_{v}	3,840	0.892	**0.841**	0.702
video	HC	-	0.265	0.247	0.193
	DML	500	0.398	0.378	0.309
	TMK	65,536	0.417	0.394	0.319
	LAMV	65,536	0.489	0.459	0.364
	VRAG	4,096	0.484	0.470	0.399
	TCA_{c}	1,024	0.570	0.553	0.473
	VVS*500* (ours)	500	0.606	0.588	0.502
	VVS*512* (ours)	512	0.608	0.590	0.505
	VVS*1024* (ours)	1,024	0.645	0.627	0.536
	VVS*3840* (ours)	3,840	**0.711**	**0.689**	**0.590**

Table 1: Benchmark on FIVR-200K. The *frame* and *video* refer to frame-level and video-level feature-based approaches. *Dim.* refers to the dimension of the basic unit for calculating similarity in each approach (i.e., frame-level approaches use multiple features of that dimension, as many as the number of all or most frames in a video, while video-level approaches use only one feature of that dimension). Only approaches that are trained from VCDB or do not require additional training are shown for a fair comparison.

	Approach	*Dim.*	CC_WEB_VIDEO			
			cc	cc^{*}	cc_{c}	cc_{c}^{*}
frame	TN	-	0.978	0.965	0.991	0.987
	DP	-	0.975	0.958	0.990	0.982
	CTE	-	**0.996**	-	-	-
	TCA_{sym}	1,024	0.982	0.962	0.992	0.981
	TCA_{f}	1,024	0.983	0.969	0.994	0.990
	TNIP	1,040	0.978	0.969	0.983	0.975
	ViSiL_{sym}	3,840	0.982	0.969	0.991	0.988
	ViSiL_{f}	3,840	0.984	0.969	0.993	0.987
	ViSiL_{v}	3,840	0.985	**0.971**	**0.996**	**0.993**
video	HC	-	0.958	-	-	-
	DML	500	0.971	0.941	0.979	0.959
	VRAG	4,096	0.971	0.952	0.980	0.967
	TCA_{c}	1,024	0.973	0.947	0.983	0.965
	VVS*500* (ours)	500	0.973	0.952	0.981	0.966
	VVS*512* (ours)	512	0.973	0.952	0.981	0.967
	VVS*1024* (ours)	1,024	0.973	0.952	0.982	0.969
	VVS*3840* (ours)	3,840	**0.975**	**0.955**	**0.984**	**0.973**

Table 2: Benchmark on CC_WEB_VIDEO. (*) refers to the evaluation of the entire dataset, and the subscript c refers to the use of cleaned annotations. All other notations and settings are identical to those presented in Table 1.

	Elim.	*Gen.*		FIVR-5K		
	DDM	TSM	TGM	DSVR	CSVR	ISVR
(a)				0.692	0.700	0.651
(b)	✓			0.715	0.725	0.672
(c)		✓		0.702	0.710	0.661
(d)			✓	0.716	0.724	0.677
(e)		✓	✓	0.719	0.726	0.680
(f)	✓	✓		0.724	0.732	0.683
(g)	✓		✓	0.738	0.746	0.698
(h)	✓	✓	✓	**0.744**	**0.752**	**0.705**

Table 3: Module-wise Ablations for VVS*3840*. *Elim.* refers to the easy distractor elimination stage, and *Gen.* refers to the suppression weight generation stage. (a) represents a baseline of the same dimension that weighs all frames equally without any of the proposed modules, (b)-(g) represent module-wise ablations, and (h) represents VVS*3840*.

in the database, while the FIVR-200K has 100 queries and 225,960 videos in the database, both of which have video-level annotations. FIVR-5K is a subset of the FIVR-200K used for ablation studies, and FIVR-200K is used for benchmarking as a large-scale video collection.

CC_WEB_VIDEO corresponds to the NDVR task, which aims to find geometrically or photometrically transformed videos. It consists of 13,129 videos in a set of 24 queries and has two types of criteria for evaluation which are divided into evaluations within each query set or within the entire video, and with the original annotation or the "cleaned" version of the annotation by (Kordopatis-Zilos et al. 2019b). The combination of these criteria provides four evaluations.

Comparison with Other Approaches

Based on the dimension C of a video-level feature V, the proposed approach is referred to as VVS_C. C is equal to that of a frame-level feature X and is determined by dimension reduction during the PCA whitening procedure. If dimension reduction is not applied, it is VVS*3840* (as used in (Kordopatis-Zilos et al. 2019b), the dimension of L_N-iMAC is 3840), and if dimension reduction is applied to match the dimension with other approaches, it is VVS*500*, VVS*512* and VVS*1024*.

Table 1 shows comparisons with previous state-of-the-art approaches on the large-scale FIVR-200K dataset. In this dataset, VVS*3840* performs approximately 25% better than the leading video-level approach in all tasks, which is close to the borderline of the frame-level state-of-the-art approaches. In addition, our approaches VVS*500*, VVS*512* and VVS*1024* are state-of-the-art regardless of whether their dimensions match or are smaller than those of other video-level approaches. This trend is similar to the performance on the CC_WEB_VIDEO in Table 2. This proves that our method is the most optimal framework between the two

	Injection Ratio	FIVR-5K		
		DSVR	CSVR	ISVR
DDM	0% - 20%	0.739	0.748	0.701
	20% - 50%	**0.744**	**0.752**	**0.705**
	50% - 80%	0.738	0.749	0.704
	80% - 100%	0.728	0.743	0.701

Table 4: Distractor Injection Ratio in DDM. This demonstrates the overall impact according to the sampling ratio of the easy distractor set in DDM.

TSM	Frame Loss $\mathcal{L}_{fr}$	FIVR-5K		
		DSVR	CSVR	ISVR
		0.742	0.749	0.702
	✓	**0.744**	**0.752**	**0.705**

Table 5: Existence of Frame Loss $\mathcal{L}_{fr}$ in TSM. This demonstrates how frame loss affects TSM.

TGM			FIVR-5K		
Init.	*Refine.*	*Hier.*	DSVR	CSVR	ISVR
Rand.	✓	✓	0.693	0.701	0.652
Const.	✓	✓	0.693	0.701	0.652
G			0.625	0.631	0.584
G	✓		0.712	0.722	0.675
G	✓	✓	**0.716**	**0.724**	**0.677**

Table 6: Structure within TGM. This demonstrates the impact of the structure within TGM. *Init.* refers to the initial state I, *Refine.* to the refinement process, and *Hier.* to the hierarchical connection. *Rand.* and *Const.* refer to situations in which the initial state is formed from a random or constant value (which is 0.5), not the rough topic representation G. To facilitate independent evaluation, the framework excludes all modules except TGM.

streams, considering that video-level approaches are essentially memory- and speed-efficient.

Ablation Studies & Analyses

Module-wise Ablations

This section covers ablation studies for each module in the proposed framework VVS$_{3840}$. As seen in Table 3, each module (b)-(d) demonstrates a significant performance increase over the baseline (a), demonstrating their value. In addition, improvements are observed even when modules are paired with one another (e)-(g), and the same is true when they are all combined (h). Moreover, in the supplementary material, by presenting further module-wise ablations of VVS$_{500}$, VVS$_{512}$ and VVS$_{1024}$, we show that all modules in our approach have a similar impact.

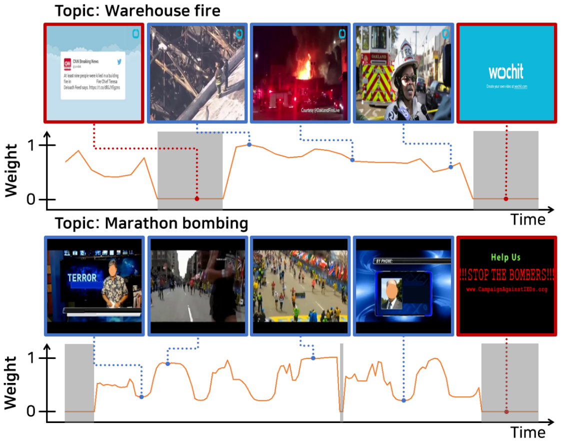

Figure 7: Qualitative Results on FIVR-5K. The orange line refers to the weights from TSM and TGM; the lower the value, the more suppressed the frame. The gray region corresponds to easy distractors eliminated by DDM, and frames that belong to this area are denoted by a red border.

Component-wise Ablations

This section covers ablation studies for components within each module of the proposed framework.

Distractor Injection Ratio in DDM. Table 4 demonstrates the effect of the sampling ratio from the easy distractor set for injection during the training phase in DDM. The model can learn slightly more cases for easy distractors compared to a lower ratio when the input length is 20–50% relative to T, leading to enhancements in the overall framework. However, when selected at a higher ratio, the proportion of frames corresponding to the distractor in a video increases excessively, which hinders optimization.

Existence of Frame Loss in TSM. To assess the impact of frame loss on TSM, Table 5 shows the outcomes of ablation when only TSM exists with no other modules. In conclusion, the frame loss allows the saliency information to be tuned, resulting in a more exact saliency label and a boost in performance.

Structure within TGM. To test the validity of the TGM structure, ablation studies for each component are shown in Table 6, and all modules other than TGM are omitted for independent evaluation of each component. First, if random or constant values are used instead of the rough topic representation G while constructing the initial state, performance deteriorates, as the model is implicitly required by relatively unclear criteria rather than explicitly guided by the topic to be well optimized. In addition, the performance gap demonstrates that even with G, the refinement process with a hierarchical connection is necessary to direct the model appropriately. Furthermore, as detailed in the supplementary material, the hierarchical connection can make the model more robust for various video lengths.

Qualitative Results

Figure 7 depicts the qualitative outcomes produced by the proposed framework. In the first example, where the topic is "Warehouse fire", it can be seen that the frames predicted by DDM as the distractors have few low-level characteristics. In addition, the fourth frame in this example is assigned a relatively low weight because no visual clues directly related to the topic appear. In the second example, where the topic is "Marathon bombing", it is shown that frames containing only text and low-level characteristics are deleted by DDM, as in the first example. Furthermore, among the remaining frames, the weights of those visually related to the topic are measured to be high, whereas the weights of the first and fourth frames, in which the scene of the event is not shown directly, are low. From these two examples, it is clear that the proposed approach achieves its intended results.

Conclusion

In this paper, we demonstrate that suppression of irrelevant frames is essential in describing an untrimmed video with long and varied content as a video-level feature. To achieve this, we present an end-to-end framework: VVS. VVS removes frames that can be clearly identified as distractors and determines the degree to which remaining frames should be suppressed based on saliency information and topic relevance. Thus, this approach is the first designed to be learned by explicit criteria, unlike previous approaches that have optimized the model implicitly. Consequently, extensive experiments proved the validity of this design and, at the same time, demonstrated that it is closest to the ideal search scenario among existing approaches due to its competitive speed and efficient memory utilization, as well as its state-of-the-art search accuracy. We hope that this work can contribute to the advancement of real-world video search systems.

Acknowledgements

This work was partly supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02067, Next Generation AI for Multi-purpose Video Search, 50%) and (RS-2023-00231894, Development of an AI Based Video Search System for Untact Content Production, 50%).

References

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Baraldi, L.; Douze, M.; Cucchiara, R.; and Jégou, H. 2018. LAMV: Learning to align and match videos with kernelized temporal layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Barrow, H. G.; Tenenbaum, J. M.; Bolles, R. C.; and Wolf, H. C. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of Image Understanding Workshop*.

Bellot, P.; and El-Bèze, M. 1999. A clustering method for information retrieval. *Technical ReportIR-0199, Laboratoire d'Informatique d'Avignon, France*.

Chou, C.-L.; Chen, H.-T.; and Lee, S.-Y. 2015. Pattern-based near-duplicate video retrieval and localization on web-scale videos. *IEEE Transactions on Multimedia*.

Douze, M., Revaud, J., Verbeek, J., Jégou, H., & Schmid, C. 2016. Circulant temporal encoding for video retrieval and temporal alignment. *International Journal of Computer Vision*.

Griffiths, A.; Luckhurst, H. C.; and Willett, P. 1986. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science*.

Gu, G.; and Ko, B. 2020. Symmetrical synthesis for deep metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jégou, H.; and Chum, O. 2012. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *European Conference on Computer Vision*.

Jiang, Y.-G.; Jiang, Y.; and Wang, J. 2014. VCDB: a large-scale database for partial copy detection in videos. In *European Conference on Computer Vision*.

Jo, W.; Lim, G.; Kim, J.; Yun, J.; and Choi, Y. 2022. Exploring the Temporal Cues to Enhance Video Retrieval on Standardized CDVA. *IEEE Access*.

JTC1/SC29/WG11/N15339, I. 2015. Call for Proposals for Compact Descriptors for Video Analysis (CDVA) – Search and Retrieval. https://mpeg.chiariglione.org/standards/exploration/compact-descriptors-video-analysis/call-proposals-compact-descriptors-video. Accessed: 2024-01-10.

Jun, H.; Ko, B.; Kim, Y.; Kim, I.; and Kim, J. 2019. Combination of multiple global descriptors for image retrieval. *arXiv preprint arXiv:1903.10663*.

Ko, B.; and Gu, G. 2020. Embedding expansion: Augmentation in embedding space for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ko, B.; Shin, M.; Gu, G.; Jun, H.; Lee, T. K.; and Kim, Y. 2019. A benchmark on tricks for large-scale image retrieval. *arXiv preprint arXiv:1907.11854*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, I. 2019a. FIVR: Fine-grained incident video retrieval. *IEEE Transactions on Multimedia*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, I. 2019b. ViSiL: Fine-grained spatio-temporal video similarity learning. In *Proceedings of IEEE Conference on Computer Vision*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, Y. 2017a. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *Proceedings of International Conference on Multimedia Modeling*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, Y. 2017b. Near-duplicate video retrieval with deep metric learning. In *Proceedings of IEEE Conference on Computer Vision Workshops*.

Kordopatis-Zilos, G.; Tzelepis, C.; Papadopoulos, S.; Kompatsiaris, I.; and Patras, I. 2022. DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval. *International Journal of Computer Vision*.

Lin, J.; Duan, L.-Y.; Wang, S.; Bai, Y.; Lou, Y.; Chandrasekhar, V.; Huang, T.; Kot, A.; and Gao, W. 2017. Hnip: Compact deep invariant representations for video matching, localization, and retrieval. *IEEE Transactions on Multimedia*.

Liu, X.; and Croft, W. B. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*.

Ng, K.; Lim, S.-N.; and Lee, G. H. 2022. VRAG: Region Attention Graphs for Content-Based Video Retrieval. *arXiv preprint arXiv:2205.09068*.

Papernot, N.; Thakurta, A.; Song, S.; Chien, S.; and Erlingsson, Ú. 2021. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Poullot, S.; Tsukatani, S.; Phuong Nguyen, A.; Jégou, H.; and Satoh, S. 2015. Temporal matching kernel with explicit feature maps. In *Proceedings of ACM International Conference on Multimedia*.

Revaud, J.; Douze, M.; Schmid, C.; and Jégou, H. 2013. Event retrieval in large video collections with circulant temporal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Shao, J.; Wen, X.; Zhao, B.; and Xue, X. 2021. Temporal context aggregation for video retrieval with contrastive learning. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision*.

Song, J.; Yang, Y.; Huang, Z.; Shen, H. T.; and Luo, J. 2013. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*.

Strzalkowski, T. 1995. Natural language information retrieval. *Information Processing & Management*.

Tan, H.-K.; Ngo, C.-W.; Hong, R.; and Chua, T.-S. 2009. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of ACM International Conference on Multimedia*.

Tolias, G.; Sicre, R.; and Jégou, H. 2016. Particular Object Retrieval With Integral Max-Pooling of CNN Activations. In *Proceedings of International Conference on Learning Representations*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wu, X.; Ngo, C.-W.; Hauptmann, A. G.; and Tan, H.-K. 2009. Real-time near-duplicate elimination for web video search with content and context. *IEEE Transactions on Multimedia*.

Zhu, M. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*.