Temporal Context Aggregation for Video Retrieval with Contrastive Learning

Jie Shao*[1,3], Xin Wen*†[2,3], Bingchen Zhao[2], and Xiangyang Xue[1]

[1]School of Computer Science, Fudan University, Shanghai, China
[2]Department of Computer Science and Technology, Tongji University, Shanghai, China
[3]ByteDance AI Lab

shaojie@fudan.edu.cn, wx99@tongji.edu.cn, zhaobc.gm@gmail.com, xyxue@fudan.edu.cn

Abstract

The current research focus on Content-Based Video Retrieval requires higher-level video representation describing the long-range semantic dependencies of relevant incidents, events, etc. However, existing methods commonly process the frames of a video as individual images or short clips, making the modeling of long-range semantic dependencies difficult. In this paper, we propose TCA (Temporal Context Aggregation for Video Retrieval), a video representation learning framework that incorporates long-range temporal information between frame-level features using the self-attention mechanism. To train it on video retrieval datasets, we propose a supervised contrastive learning method that performs automatic hard negative mining and utilizes the memory bank mechanism to increase the capacity of negative samples. Extensive experiments are conducted on multiple video retrieval tasks, such as CC_WEB_VIDEO, FIVR-200K, and EVVE. The proposed method shows a significant performance advantage ($\sim 17\%$ *mAP on FIVR-200K) over state-of-the-art methods with video-level features, and deliver competitive results with 22x faster inference time comparing with frame-level features.*

Figure 1: Example query describing the crash of a hawker hunter at Shoreham airport and its challenging distractors retrieved from the FIVR-200K [30] dataset. As the scene of the aircraft in the sky takes the majority of the video, the vital information about the crash (with fewer frames) is covered up, thus the mistakenly retrieved videos share similar scenes, but describe totally different events.

1. Introduction

We address the task of Content-Based Video Retrieval. The research focus on Content-Based Video Retrieval has shifted from Near-Duplicate Video Retrieval (NDVR) [61, 25] to Fine-grained Incident Video Retrieval [30], Event-based Video Retrieval [45], *etc*. Different from NDVR, these tasks are more challenging in terms that they require higher-level representation describing the long-range semantic dependencies of relevant incidents, events, *etc*.

The central task of Contend-Based Video Retrieval is to predict the similarity between video pairs. Current approaches mainly follow two schemes: to compute the similarity using video-level representations (first scheme) or frame-level representations (second scheme). For methods using video-level representations, early studies typically employ code books [6, 32, 35] or hashing functions [51, 52] to form video representations, while later approach (Deep Metric Learning [33]) is introduced to generate video representations by aggregating the pre-extracted frame-level representations. In contrast, the approaches following the second scheme typically extract frame-level representations to compute frame-to-frame similarities, which are then used to obtain video-level similarities [9, 36, 31, 54]. With more elaborate similarity measurements, they typically outperform those methods with the first scheme.

For both schemes, the frames of a video are commonly processed as individual images or short clips, making the

*Both authors contributed equally to this work.
†Work done while Xin Wen was an intern at ByteDance AI Lab.

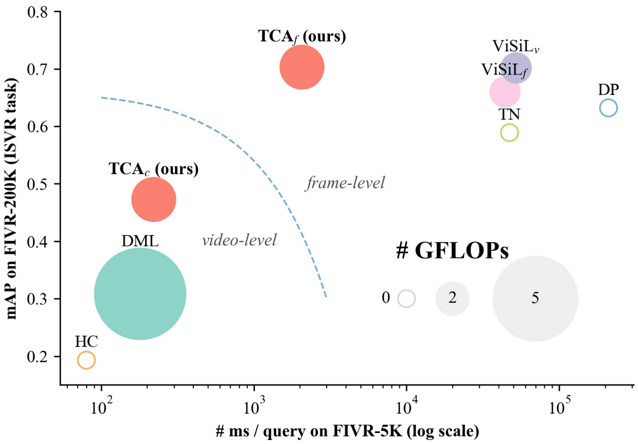


Figure 2: Video Retrieval performance comparison on ISVR task of FIVR [30] in terms of mAP, inference time, and computational cost of the model (ISVR is the most complete and hard task of FIVR). The proposed approach achieves the best trade-off between performance and efficiency with both video-level and frame-level features against state-of-the-art methods. (*Best viewed in color*)

modeling of long-range semantic dependencies difficult. As the visual scene of videos can be redundant (such as scenery shots or B-rolls), potentially unnecessary visual data may dominate the video representation, and mislead the model to retrieve negative samples sharing similar scenes, as the example shown in Fig. 1. Motivated by the effectiveness of the self-attention mechanism in capturing long-range dependencies [57], we propose to incorporate temporal information between frame-level features (*i.e.*, temporal context aggregation) using the self-attention mechanism to better model the long-range semantic dependencies, helping the model focus on more informative frames, thus obtaining more relevant and robust features.

To supervise the optimization of video retrieval models, current state-of-the-art methods [33, 31] commonly perform pair-wise optimization with triplet loss [60]. However, the relation that triplets can cover is limited, and the performance of triplet loss is highly subject to the time-consuming hard-negative sampling process [50]. Inspired by the recent success of contrastive learning on self-supervised learning [17, 7] and the nature of video retrieval datasets that rich negative samples are readily available, we propose a supervised contrastive learning method for video retrieval. With the help of a shared memory bank, large quantities of negative samples are utilized efficiently with no need for manual hard-negative sampling. Furthermore, by conducting gradient analysis, we show that our proposed method has the property of automatic hard-negative mining which could greatly improve the final performance.

Extensive experiments are conducted on multi video retrieval datasets, such as CC_WEB_VIDEO [61], FIVR [30], and EVVE [45]. In comparison with previous methods, as shown in Fig. 2, the proposed method shows a significant performance advantage (*e.g.*, $\sim 17\%$ mAP on FIVR-200K) over state-of-the-art methods with video-level features, and deliver competitive results with 22x faster inference time comparing with methods using frame-level features.

2. Related Work

Frame Feature Representation. Early approaches employed handcrafted features including the Scale-Invariant Feature Transform (SIFT) features [26, 38, 61], the Speeded-Up Robust Features (SURF) [5, 9], Colour Histograms in HSV space [16, 27, 52], and Local Binary Patterns (LBP) [66, 48, 62], *etc.* Recently, Deep Convolutional Neural Networks (CNNs) have proved to be versatile representation tools in recent approaches. The application of Maximum Activation of Convolutions (MAC) and its variants [44, 68, 43, 56, 67, 46, 14], which extract frame descriptors from activations of a pre-trained CNN model, have achieved great success in both fine-grained image retrieval and video retrieval tasks [14, 32, 34, 33, 31]. Besides variants of MAC, Sum-Pooled Convolutional features (SPoC) [3] and Generalized Mean (GeM) [15] pooling are also considerable counterparts.

Video Feature Aggregation. Typically, the video feature aggregation paradigm can be divided into two categories: (1) local feature aggregation models [10, 49, 42, 24] which are derived from traditional local image feature aggregation models, and (2) sequence models [20, 8, 11, 13, 57, 64] that model the temporal order of the video representation. Popular local feature aggregation models include Bag-of-Words [10, 49], Fisher Vector [42], and Vector of Locally Aggregated Descriptors (VLAD) [24], of which the unsupervised learning of a visual code book is required. The NetVLAD [1] transfers VLAD into a differential version, and the clusters are tuned via back-propagation instead of k-means clustering. In terms of the sequence models, the Long Short-Term Memory (LSTM) [20] and Gated Recurrent Unit (GRU) [8] are commonly used for video re-localization and copy detection [13, 22]. Besides, self-attention mechanism also shows success in video classification [59] and object detection [21].

Contrastive Learning. Contrastive learning has become the common training paradigm of recent self-supervised learning works [40, 19, 55, 17, 7, 65], in which the positive and negative sample pairs are constructed with a pretext task in advance, and the model tries to distinguish the positive sample from massive randomly sampled negative samples in a classification manner. The contrastive loss typically performs better in general than triplet loss for representation learning [7] which can only handle one positive and negative at a time. The core of the effectiveness of contrastive learning is the use of rich negative samples [55], one approach is to sample them from a shared memory bank [63], and [17] replaced the bank with a queue and used a moving-averaged encoder to build a larger and consistent dictionary on-the-fly.

3. Method

In this section, we first define the problem setting (Section 3.1) and describe the frame-level feature extraction step (Section 3.2). Then, we demonstrate the temporal context aggregation module (Section 3.3) and the contrastive learning method based on pair-wise video labels (Section 3.4), then conduct further analysis on the gradients of the loss function (Section 3.5). And last, we discuss the similarity measure of video-level and frame-level video descriptors (Section 3.6).

3.1. Problem Setting

We address the problem of video representation learning for Near-Duplicate Video Retrieval (NDVR), Fine-grained Incident Video Retrieval (FIVR), and Event Video Retrieval (EVR) tasks. In our setting, the dataset is two-split: the *core* and *distractor*. The core subset contains pair-wise labels describing which two videos are similar (near duplicate, complementary scene, same event, *etc*.). And the distractor subset contain large quantities of negative samples to make the retrieval task more challenging.

We only consider the RGB data of the videos. Given raw pixels ($\mathbf{x}_r \in \mathbb{R}^{m \times n \times f}$), a video is encoded into a sequence of frame-level descriptors ($\mathbf{x}_f \in \mathbb{R}^{d \times f}$) or a compact video-level descriptor ($\mathbf{x}_v \in \mathbb{R}^d$). Take the similarity function as $\mathrm{sim}(\cdot, \cdot)$, the similarity of two video descriptors $\mathbf{x}_1, \mathbf{x}_2$ can be denoted as $\mathrm{sim}(\mathbf{x}_1, \mathbf{x}_2)$. Given these, our task is to optimize the embedding function $f(\cdot)$, such that $\mathrm{sim}\left(f\left(\mathbf{x}_1\right), f\left(\mathbf{x}_2\right)\right)$ is maximized if $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar videos, and minimized otherwise. The embedding function $f(\cdot)$ typically takes a video-level descriptor $\mathbf{x} \in \mathbb{R}^d$ and returns an embedding $f(\mathbf{x}) \in \mathbb{R}^k$, in which $k \ll d$. However, in our setting, $f(\cdot)$ is a temporal context aggregation modeling module, thus frame-level descriptors $\mathbf{x} \in \mathbb{R}^{d \times f}$ are taken as input, and the output can be either aggregated video-level descriptor ($f(\mathbf{x}) \in \mathbb{R}^d$) or refined frame-level descriptors ($f(\mathbf{x}) \in \mathbb{R}^{d \times f}$).

3.2. Feature Extraction

According to the results reported in [31] (Table 2), we select iMAC [14] and modified L$_3$-iMAC [31] (called L$_3$-iRMAC) as our benchmark frame-level feature extraction methods. Given a pre-trained CNN network with K convolutional layers, K feature maps $\mathcal{M}^k \in \mathbb{R}^{n_d^k \times n_d^k \times c^k} (k = 1, \ldots, K)$ are generated, where $n_d^k \times n_d^k$ is the dimension of each feature map of the k^{th} layer, and c^k is the total number of channels.

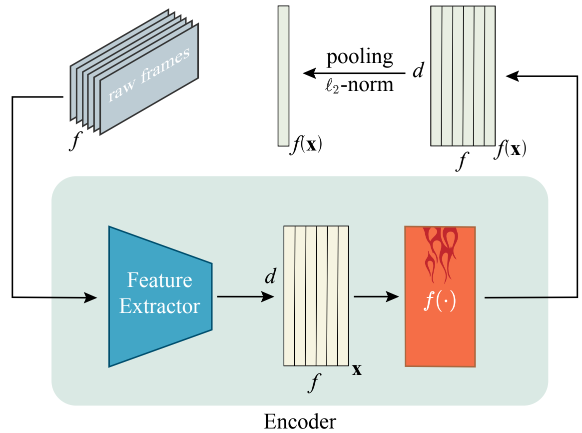


Figure 3: **Feature encoding pipeline.** Raw frames are fed to the feature extractor to extract the frame-level video descriptor $\mathbf{x}$. Then the self-attention mechanism is applied to perform temporal context aggregation on the input, and generate refined frame-level descriptors $f(\mathbf{x})$. They can also be compressed into a video-level descriptor by applying average pooling and ℓ_2-normalization.

For iMAC feature, the maximum value of every channel of each layer is extracted to generate K feature maps $\mathcal{M}^k \in \mathbb{R}^{c^k}$, as formulated in Eq. 1:

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

where v^k is a c^k-dimensional vector that is derived from max pooling on each channel of the feature map $\mathcal{M}^k$.

Max pooling with different kernel size and stride are applied to every channel of different layers to generate K feature maps $\mathcal{M}^k \in \mathbb{R}^{3 \times 3 \times c^k}$ in the original L$_3$-iMAC feature. Unlike its setting, we then follow the tradition of R-MAC [56] to sum the 3×3 feature maps together, then apply ℓ_2-normalization on each channel to form a feature map $\mathcal{M}^k \in \mathbb{R}^{c^k}$. This presents a trade-off between the preservation of fine-trained spatial information and low feature dimensionality (equal to iMAC), we denote this approach as L$_3$-iRMAC.

For both iMAC and L$_3$-iRMAC, all layer vectors are concatenated to a single descriptor after extraction, then PCA is applied to perform whitening and dimensionality reduction following the common practice [23, 31], finally ℓ_2-normalization is applied on each channel, resulting in a compact frame-level descriptor $\mathbf{x} \in \mathbb{R}^{d \times f}$.

3.3. Temporal Context Aggregation

We adopt the Transformer [57] model for temporal context aggregation. Following the setting of [13, 64], only the encoder structure of the Transformer is used. With the parameter matrices written as W^Q, W^K, W^V, the entire video descriptor $\mathbf{x} \in \mathbb{R}^{d \times f}$ is first encoded into Query Q, Key

K and Value V by three different linear transformations: $Q = \mathbf{x}^\top W^Q$, $K = \mathbf{x}^\top W^K$ and $V = \mathbf{x}^\top W^V$. This is further calculated by the self-attention layer as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V \,. \tag{2}$$

The result is then taken to the LayerNorm layer [2] and Feed Forward Layer [57] to get the output of the Transformer encoder, *i.e.*, $f_{\text{Transformer}}(\mathbf{x}) \in \mathbb{R}^{d \times f}$. The multi-head attention mechanism is also used.

With the help of the self-attention mechanism, Transformer is effective at modeling long-term dependencies within the frame sequence. Although the encoded feature keeps the same shape as the input, the contextual information within a longer range of each frame-level descriptor is incorporated. Apart from the frame-level descriptor, by simply averaging the encoded frame-level video descriptors along the time axis, we can also get the compact video-level representation $\overline{f}(\mathbf{x}) \in \mathbb{R}^d$.

3.4. Contrastive Learning

If we denote $\mathbf{w}_a, \mathbf{w}_p, \mathbf{w}_n^j (j = 1, 2, \ldots, N-1)$ as the video-level representation before applying normalization of the anchor, positive, negative examples, we get the similarity scores by: $s_p = \mathbf{w}_a^\top \mathbf{w}_p / (\|\mathbf{w}_a\| \, \|\mathbf{w}_p\|)$ and $s_n^j = \mathbf{w}_a^\top \mathbf{w}_n^j / \left(\|\mathbf{w}_a\| \, \|\mathbf{w}_n^j\|\right)$. Then the InfoNCE [40] loss is written as:

$$\mathcal{L}_{\text{nce}} = -\log \frac{\exp\left(s_p/\tau\right)}{\exp\left(s_p\right) + \sum_{j=1}^{N-1} \exp\left(s_n^j/\tau\right)} \,, \tag{3}$$

where τ is a temperature hyper-parameter [63]. To utilize more negative samples for better performance, we borrow the idea of the memory bank from [63]. For each batch, we take one positive pair from the core dataset and randomly sample n negative samples from the distractors, then the compact video-level descriptors are generated with a shared encoder. The negative samples of all batches from all GPUs (k batches in total) are concatenated together to form the memory bank. We compare the similarity of the anchor sample against the positive sample and all negatives in the memory bank, resulting in 1 s_p and $N = kn$ s_n. Then the loss can be calculated in a classification manner. The momentum mechanism [17] is not adopted as we did not see any improvement in experiments. Besides the InfoNCE loss, the recent proposed Circle Loss [53] is also considered:

$$\mathcal{L}_{\text{circle}} = -\log \frac{\exp(\gamma \alpha_p (s_p - \Delta_p))}{\exp(\gamma \alpha_p (s_p - \Delta_p)) + \sum_{j=1}^{N-1} \exp(\gamma \alpha_n^j (s_n^j - \Delta_n))} \tag{4}$$

where γ is the scale factor (equivalent with the parameter τ in Eq. 3), and m is the relaxation margin. $\alpha_p = [1 + m - s_p]_+ , \alpha_n^j = \left[s_n^j + m\right]_+ , \Delta_p = 1 - m, \Delta_n = m$. Compared with the InfoNCE loss, the Circle loss optimizes s_p and s_n separately with adaptive penalty strength and adds within-class and between-class margins.

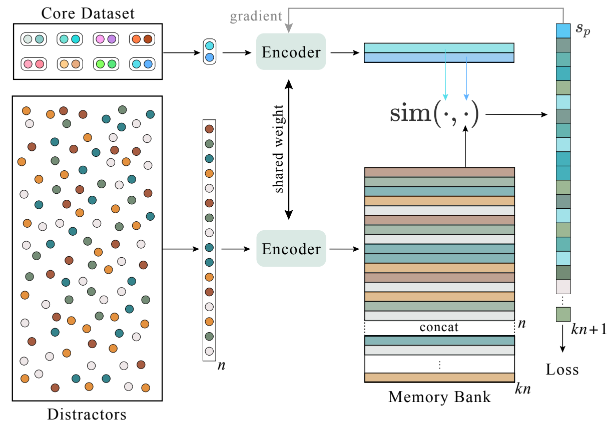


Figure 4: **Learning representation with pair-wise labels.** For each batch, we take one positive pair from the core dataset and randomly sample n negative samples from the distractors, then the video-level descriptors are generated with a shared encoder. The negative samples of all batches and all GPUs are concatenated together to form the memory bank. We compare the similarity of the anchor against the positive sample and all negatives in the memory bank, resulting in 1 s_p and kn s_n. Then the loss can be calculated in a classification manner following Eq. 3 and Eq. 4.

3.5. One Step Further on the Gradients

In the recent work of Khosla *et al.* [28], the proposed batch contrastive loss is proved to focus on the hard positives and negatives automatically with the help of feature normalization by conducting gradient analysis, we further reveal that this is the common property of Softmax loss and its variants when combined with feature normalization. For simplicity, we analyze the gradients of Softmax loss, the origin of both InfoNCE loss and Circle loss:

$$\mathcal{L}_{\text{softmax}} = -\log \frac{\exp\left(s_p\right)}{\exp\left(s_p\right) + \sum_{j=1}^{N-1} \exp\left(s_n^j\right)} \,, \tag{5}$$

the notation is as aforementioned. Here we show that easy negatives contribute the gradient weakly while hard negatives contribute greater. With the notations declared in Section 3.4, we denote the normalized video-level representation as $\mathbf{z}_* = \mathbf{w}_* / \|\mathbf{w}_*\|$, then the gradients of Eq. 5 with respect to $\mathbf{w}_a$ is:

$$\begin{aligned}
&\frac{\partial \mathcal{L}_{\text{softmax}}}{\partial \mathbf{w}_a} = \frac{\partial \mathbf{z}_a}{\partial \mathbf{w}_a} \cdot \frac{\partial \mathcal{L}_{\text{softmax}}}{\partial \mathbf{z}_a} \\
&= \frac{1}{\|\mathbf{w}_a\|} \left(\mathbf{I} - \mathbf{z}_a \mathbf{z}_a^\top\right) \left[\left(\sigma(\mathbf{s})_p - 1\right) \mathbf{z}_p + \sum_{j=1}^{N-1} \sigma(\mathbf{s})_n^j \mathbf{z}_n^j \right] \\
&\propto \overbrace{(1 - \sigma(\mathbf{s})_p)[(\mathbf{z}_a^\top \mathbf{z}_p)\mathbf{z}_a - \mathbf{z}_p]}^{\text{positive}} + \underbrace{\sum_{j=1}^{N-1} \sigma(\mathbf{s})_n^j [\mathbf{z}_n^j - (\mathbf{z}_a^\top \mathbf{z}_n^j)\mathbf{z}_a]}_{\text{negatives}} \,,
\end{aligned} \tag{6}$$

where $\sigma(\mathbf{s})_p = \exp(s_p) \Big/ \left[\exp(s_p) + \sum_{j=1}^{N-1} \exp\left(s_n^j\right)\right]$, and $\sigma(\mathbf{s})_n^j = \exp\left(s_n^j\right) \Big/ \left[\exp(s_p) + \sum_{j=1}^{N-1} \exp\left(s_n^j\right)\right]$ following the common notation of the softmax function. For an easy negative, the similarity between it and the anchor is close to -1, thus $\mathbf{z}_a^\top \mathbf{z}_n^j \approx -1$, and therefore

$$\sigma(\mathbf{s})_n^j \left\| \left(\mathbf{z}_n^j - \left(\mathbf{z}_a^\top \mathbf{z}_n^j\right) \mathbf{z}_a\right) \right\| = \sigma(\mathbf{s})_n^j \sqrt{1 - \left(\mathbf{z}_a^\top \mathbf{z}_n^j\right)^2} \approx 0\,. \tag{7}$$

And for a hard negative, $\mathbf{z}_a^\top \mathbf{z}_n^j \approx 0$[1], and $\sigma(\mathbf{s})_n^j$ is moderate, thus the above equation is greater than 0, and its contribution to the gradient of the loss function is greater. Former research only explained it intuitively that features with shorter amplitudes often represent categories that are more difficult to distinguish, and applying feature normalization would divide harder examples with a smaller value (the amplitude), thus getting relatively larger gradients [58], however, we prove this property for the first time by conducting gradient analysis. The derivation process of Eq. 3 and Eq. 4 are alike. Comparing with the commonly used Triplet loss in video retrieval tasks [33, 31] which requires computationally expensive hard negative mining, the proposed method based on contrastive learning takes advantage of the nature of softmax-based loss when combined with feature normalization to perform hard negative mining automatically, and use the memory bank mechanism to increase the capacity of negative samples, which greatly improves the training efficiency and effect.

3.6. Similarity Measure

To save the computation and memory cost, at the training stage, all feature aggregation models are trained with the output as ℓ_2-normalized video-level descriptors ($f(\mathbf{x}) \in \mathbb{R}^d$), thus the similarity between video pairs is simply calculated by dot product. Besides, for the sequence aggregation models, refined frame-level video descriptors ($f(\mathbf{x}) \in \mathbb{R}^{d\times f}$) can also be easily extracted before applying average pooling along the time axis. Following the setting in [31], at the evaluation stage, we also use chamfer similarity to calculate the similarity between two frame-level video descriptors. Denote the representation of two videos as $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^\top$, $\mathbf{y} = [y_0, y_1, \ldots, y_{m-1}]^\top$, where $x_i, y_j \in \mathbb{R}^d$, the chamfer similarity between them is:

$$\mathrm{sim}_f(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=0}^{n-1} \max_j x_i y_j^\top\,, \tag{8}$$

and the symmetric version:

$$\mathrm{sim}_{sym}(\mathbf{x}, \mathbf{y}) = (\mathrm{sim}_f(\mathbf{x}, \mathbf{y}) + \mathrm{sim}_f(\mathbf{y}, \mathbf{x}))/2\,. \tag{9}$$

[1]This represents the majority of hard negatives, and if the similarity is close to 1, it is too hard and may cause the model to collapse, or due to wrong annotation.

Note that this approach (chamfer similarity) seems to be inconsistent with the training target (cosine similarity), where the frame-level video descriptors are averaged into a compact representation and the similarity is calculated with dot product. However, the similarity calculation process of the compact video descriptors can be written as:

$$\begin{aligned} \mathrm{sim}_{cos}(\mathbf{x}, \mathbf{y}) &= \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i\right) \left(\frac{1}{m} \sum_{j=0}^{m-1} y_j\right)^\top \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{m} \sum_{j=0}^{m-1} x_i y_j^\top\,. \end{aligned} \tag{10}$$

Therefore, given frame-level features, chamfer similarity averages the maximum value of each row of the video-video similarity matrix, while cosine similarity averages the mean value of each row. It is obvious that $\mathrm{sim}_{cos}(\mathbf{x}, \mathbf{y}) \le \mathrm{sim}_f(\mathbf{x}, \mathbf{y})$, therefore, by optimizing the cosine similarity, we are optimizing the lower-bound of the chamfer similarity. As only the compact video-level feature is required, both time and space complexity are greatly reduced as cosine similarity is much computational efficient.

4. Experiments

4.1. Experiment Setting

We evaluate the proposed approach on three video retrieval tasks, namely Near-Duplicate Video Retrieval (NDVR), Fine-grained Incident Video Retrieval (FIVR), and Event Video Retrieval (EVR). In all cases, we report the mean Average Precision (mAP).

Training Dataset. We leverage the **VCDB** [25] dataset as the training dataset. The core dataset of VCDB has 528 query videos and 6,139 positive pairs, and the distractor dataset has 100,000 distractor videos, of which we successfully downloaded 99,181 of them.

Evaluation Dataset. For models trained on the VCDB dataset, we test them on the **CC_WEB_VIDEO** [61] dataset for NDVR task, **FIVR-200K** for FIVR task and **EVVE** [45] for EVR task. For a quick comparison of the different variants, the **FIVR-5K** dataset as in [31] is also used. The CC_WEB_VIDEO dataset contains 24 query videos and 13,129 labeled videos; The FIVR-200K dataset includes 225,960 videos and 100 queries, it consists of three different fine-grained video retrieval tasks: (1) Duplicate Scene Video Retrieval, (2) Complementary Scene Video Retrieval and (3) Incident Scene Video Retrieval; The EVVE dataset is designed for the EVR task, it consists of 2,375 videos and 620 queries.

Implementation Details. For feature extraction, we extract one frame per second for all videos. For all retrieval tasks, we extract the frame-level features following the scheme in Section 3.2. The intermediate features are all extracted from the output of four residual blocks of ResNet-

Model	DSVR	CSVR	ISVR
NetVLAD	0.513	0.494	0.412
LSTM	0.505	0.483	0.400
GRU	0.515	0.495	0.415
Transformer	**0.551**	**0.532**	**0.454**

(a) **Model** (mAP on FIVR-5K)

Feature	DSVR	CSVR	ISVR
iMAC	0.547	0.526	0.447
L_3-iRMAC	**0.570**	**0.553**	**0.473**

(b) **Feature** (mAP on FIVR-200K)

Loss	τ/γ	DSVR	CSVR	ISVR
InfoNCE	0.07	0.493	0.473	0.394
InfoNCE	$1/256$	0.566	0.548	0.468
Circle	256	**0.570**	**0.553**	**0.473**

(c) **Loss function** (mAP on FIVR-200K)

Method	Bank Size	DSVR	CSVR	ISVR
triplet	-	0.510	0.509	0.455
ours	256	0.605	0.615	0.575
ours	4096	0.609	**0.617**	**0.578**
ours	65536	**0.611**	**0.617**	0.574

(d) **Bank size** (mAP on FIVR-5K)

Momentum	DSVR	CSVR	ISVR
0 (bank)	**0.609**	**0.617**	**0.578**
0.1	0.606	0.612	0.569
0.9	0.605	0.611	0.568
0.99	0.602	0.606	0.561
0.999	0.581	0.577	0.520

(e) **Momentum** (mAP on FIVR-5K)

Similarity Measure	DSVR	CSVR	ISVR
cosine	0.609	0.617	0.578
chamfer	**0.844**	**0.834**	**0.763**
symm. chamfer	0.763	0.766	0.711
chamfer+comparator	0.726	0.735	0.701

(f) **Similarity Measure** (mAP on FIVR-5K)

Table 1: **Ablations on FIVR about:** (a): Temporal context aggregation methods; (b): Frame feature representations; (c): Loss functions for contrastive learning ($\gamma = 1/\tau$); (d) Size of the memory bank; (e) Momentum parameter of the queue of MoCo [17], degenerate to memory bank when set to 0; (f) Similarity measures (video-level and frame-level), comparator: the comparator network used in ViSiL$_v$ [31], with original parameters retained.

50 [18]. PCA trained on 997,090 randomly sampled frame-level descriptors from VCDB is applied to both iMAC and L_3-iRMAC features to perform whitening and reduce its dimension from 3840 to 1024. Finally, ℓ_2-normalization is applied.

For the Transformer model, it is implemented with one single layer, eight attention heads, dropout_rate set to 0.5, and the dimension of the feed-forward layer set to 2048.

During training, all videos are padded to 64 frames (if longer, a random segment with a length of 64 is extracted), and the full video is used in the evaluation stage. Adam [29] is adopted as the optimizer, with the initial learning rate set to 10^{-5}, and cosine annealing learning rate scheduler [37] is used. The model is trained with batch size 64 for 40 epochs, and 16×64 negative samples sampled from the distractors are sent to the memory bank each batch, with a single device with four Tesla-V100-SXM2-32GB GPUs, the size of the memory bank is equal to 4096. The code is implemented with PyTorch [41], and distributed training is implemented with Horovod [47].

4.2. Ablation Study

Models for Temporal Context Aggregation. In Table 1a, we compare the Transformer with prior temporal context aggregation approaches, *i.e.*, NetVLAD [1], LSTM [20] and GRU [8]. All models are trained on VCDB dataset with iMAC feature and evaluated on all three tasks of FIVR-5K, and dot product is used for similarity calculation for both train and evaluation. The classic recurrent models (LSTM, GRU) do not show advantage against NetVLAD. However, with the help of self-attention mechanism, the Transformer model demonstrate excellent performance gain in almost all tasks, indicating its strong ability of long-term temporal dependency modeling.

Frame Feature Representation. We evaluate the iMAC and L_3-iRMAC feature on the FIVR-200K dataset with cosine similarity, as shown in Table 1b. With more local spatial information leveraged, L_3-iRMAC show consistent improvement against iMAC.

Loss function for contrastive learning. We present the comparison of loss functions for contrastive learning in Table 1c. The InfoNCE loss show notable inferiority compared with Circle with default parameters $\tau = 0.07, \gamma = 256, m = 0.25$. By adjusting the sensitive temperature parameter τ (set to $1/256$, equivalent with $\gamma = 256$ in Circle loss), it still shows around 0.5% less mAP.

Size of the Memory Bank. In Table 1d, we present the comparison of different sizes of the memory bank. It is observed that a larger memory bank convey consistent performance gain, indicating the efficiency of utilizing large quantities of negative samples. Besides, we compare our approach against the commonly used triplet based approach with hard negative mining [33] (without bank). The training process of the triplet-based scheme is extremely time-consuming (5 epochs, 5 hours on 32 GPUs), yet still show around 10% lower mAP compared with the baseline (40 epochs, 15 minutes on 4 GPUs), indicating that compared with learning from hard negatives, to utilize a large number of randomly sampled negative samples is not only more efficient, but also more effective.

Momentum Parameter. In Table 1e, we present the ablation on momentum parameter of the modified MoCo [17]-

like approach, where a large queue is maintained to store the negative samples and the weight of the model is updated in a moving averaged manner. We experimented with different momentum ranging from 0.1 to 0.999 (with queue length set to 65536), but none of them show better performance than the baseline approach as reported in Table 1d, we argue that the momentum mechanism is a compromise for larger memory. as the memory bank is big enough in our case, the momentum mechanism is not needed.

Similarity Measure. We evaluate the video-level features with cosine similarity, and frame-level features following the setting of ViSiL [31], *i.e.*, chamfer similarity, symmetric chamfer similarity, and chamfer similarity with similarity comparator (the weights are kept as provided by the authors). Table 1f presents the results on FIVR-5K dataset. Interestingly, the frame-level similarity calculation approach outperforms the video-level approach by a large margin, indicating that frame-level comparison is important for fine-grained similarity calculation between videos. Besides, the comparator network does not show as good results as reported, we argue that this may be due to the bias between features.

Next, we only consider the Transformer model trained with L_3-iRMAC feature and Circle loss in the following experiments, denoted as TCA (Temporal Context Encoding for Video Retrieval). With different similarity measures, all four approaches are denoted as TCA_c (cosine), TCA_f (chamfer), TCA_{sym} (symmetric-chamfer), TCA_v (video comparator) for simplicity.

4.3. Comparison Against State-of-the-art

Near-duplicate Video Retrieval. We first compare TCA against state-of-the-art methods on several versions of CC_WEB_VIDEO [61]. The benchmark approaches are Deep Metric Learning (DML) [33], the Circulant Temporal Encoding (CTE) [45], and Fine-grained Spatio-Temporal Video Similarity Learning (ViSiL), we report the best results of the original paper. As listed in Table 2, we report state-of-the-art results on all tasks with video-level features, and competitive results against ViSiL_v with refined frame-level features. To emphasize again, our target is to learn a good video representation, and the similarity calculation stage is expected to be as simple and efficient as possible, therefore, it is fairer to compare TCA_f with ViSiL_f, as they hold akin similarity calculation approach.

Fine-grained Incident Video Retrieval. We evaluate TCA against state-of-the-art methods on FIVR-200K [30]. We report the best results reported in the original paper of DML [33], Hashing Codes (HC) [52], ViSiL [31], and their re-implemented DP [9] and TN [54]. As shown in Table 3, the proposed method shows a clear performance advantage over state-of-the-art methods with video-level features (TCA_c), and deliver competitive results with frame-level features (TCA_f). Compared with ViSiL_f, we show a clear performance advantage even with a more compact frame-level feature and simpler frame-frame similarity measure.

A more comprehensive comparison on performance is given in Fig. 2. The proposed approach achieves the best trade-off between performance and efficiency with both video-level and frame-level features against state-of-the-art methods. When compared with ViSiL_v, we show competitive results with about 22x faster inference time. Interestingly, our method slightly outperforms ViSiL_v in ISVR task, indicating that by conducting temporal context aggregation, our model might show an advantage in extracting semantic information.

	Method	CC_WEB_VIDEO			
		cc_web	cc_web*	cc_web$_c$	cc_web$_c$*
Video-level	DML [33]	0.971	0.941	0.979	0.959
	TCA_c	**0.973**	**0.947**	**0.983**	**0.965**
Frame-level	CTE [45]	**0.996**	-	-	-
	ViSiL_f [31]	0.984	0.969	0.993	0.987
	ViSiL_{sym} [31]	0.982	0.969	0.991	0.988
	ViSiL_v [31]	0.985	**0.971**	**0.996**	**0.993**
	TCA_f	0.983	0.969	0.994	0.990
	TCA_{sym}	0.982	0.962	0.992	0.981

Table 2: **mAP on 4 versions of CC_WEB_VIDEO.** Following the setting in ViSiL [31], (*) denotes evaluation on the entire dataset, and subscript c denotes using the cleaned version of the annotations.

	Method	FIVR-200K			EVVE
		DSVR	CSVR	ISVR	
Video-level	DML [33]	0.398	0.378	0.309	-
	HC [52]	0.265	0.247	0.193	-
	LAMV+QE [4]	-	-	-	0.587
	TCA_c	**0.570**	**0.553**	**0.473**	**0.598**
Frame-level	DP [9]	0.775	0.740	0.632	-
	TN [54]	0.724	0.699	0.589	-
	ViSiL_f [31]	0.843	0.797	0.660	0.597
	ViSiL_{sym} [31]	0.833	0.792	0.654	0.616
	ViSiL_v [31]	**0.892**	**0.841**	0.702	0.623
	TCA_f	0.877	0.830	**0.703**	0.603
	TCA_{sym}	0.728	0.698	0.592	**0.630**

Table 3: **mAP on FIVR-200K and EVVE.** The proposed approach achieves the best trade-off between performance and efficiency with both video-level and frame-level features against state-of-the-art methods.

Event Video Retrieval. For EVR, we compare TCA with Learning to Align and Match Videos (LAMV) [4] with

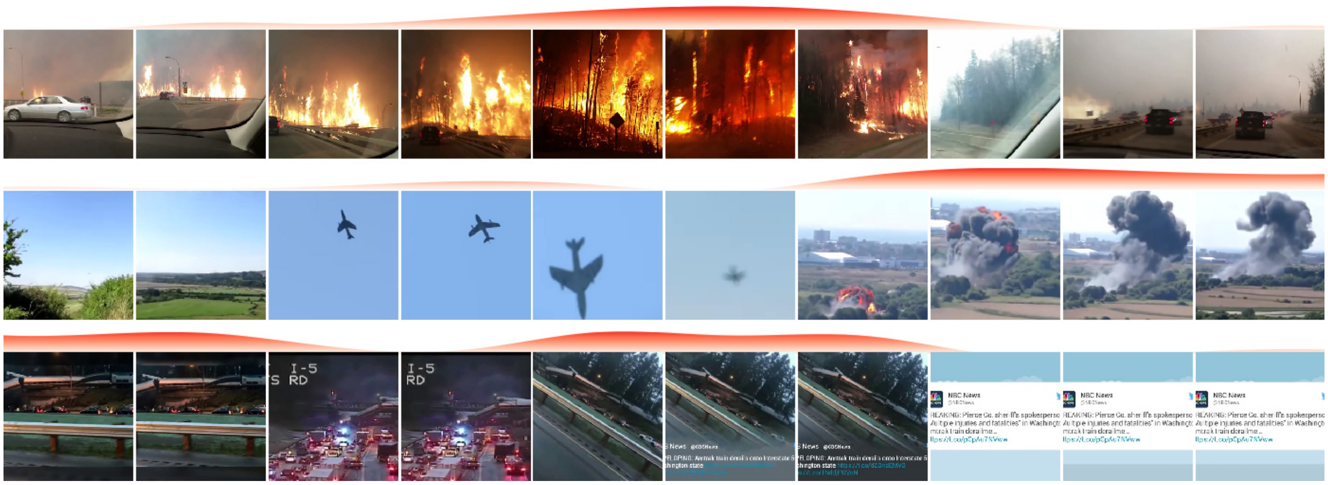
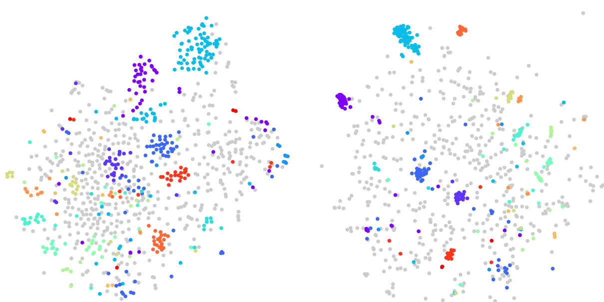


Figure 5: **Visualization of average attention weight (response) of example videos in FIVR.** The weights are normalized and interpolated for better visualization, and darker color indicates higher average response of the corresponding frame. Each case tends to focus on salient and informative frames: video #1 focuses on key segments about the fire; video #2 has a higher focus on the explosion segment; and video #3 selectively ignores the meaningless ending.



(a) DML [33] (b) Ours (TCA_c)

Figure 6: **Visualization of video-level features on a subset of FIVR-5K with t-SNE.** Each color represents samples corresponding to one single query, and distractors are colored with faded gray. Both our method and DML are trained on VCDB [25] dataset. (*Best viewed in color*)

Average Query Expansion (AQE) [12] and ViSiL [31] on EVVE [45]. We report the results of LAMV from the original paper, and the re-evaluated ViSiL (the reported results are evaluated on incomplete data). As shown in Table 3, TCA_{sym} achieves the best result. Surprisingly, our video-level feature version TCA_c also report notable results, this may indicate that the temporal information and fine-grained spatial information are not necessary for event video retrieval task.

4.4. Qualitative Results

We demonstrate the distribution of video-level features on a randomly sampled subset of FIVR-5K with t-SNE [39] in Fig. 6. Compared with DML, the clusters formed by relevant videos in the refined feature space obtained by our approach are more compact, and the distractors are better separated; To better understand the effect of the self-attention mechanism, we visualize the average attention weight (response) of three example videos in Fig. 5. The self-attention mechanism helps expand the vision of the model from separate frames or clips to almost the whole video, and conveys better modeling of long-range semantic dependencies within the video. As a result, informative frames describing key moments of the event get higher response, and the redundant frames are suppressed.

5. Conclusion

In this paper, we present TCA, a video representation learning network that incorporates temporal-information between frame-level features using self-attention mechanism to help model long-range semantic dependencies for video retrieval. To train it on video retrieval datasets, we propose a supervised contrastive learning method. With the help of a shared memory bank, large quantities of negative samples are utilized efficiently with no need for manual hard-negative sampling. Furthermore, by conducting gradient analysis, we show that our proposed method has the property of automatic hard-negative mining which could greatly improve the final model performance. Extensive experiments are conducted on multi video retrieval tasks, and the proposed method achieves the best trade-off between performance and efficiency with both video-level and frame-level features against state-of-the-art methods.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[4] Lorenzo Baraldi, Matthijs Douze, Rita Cucchiara, and Hervé Jégou. Lamv: Learning to align and match videos with kernelized temporal layers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7804–7813, 2018.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Yang Cai, Linjun Yang, Wei Ping, Fei Wang, Tao Mei, Xian-Sheng Hua, and Shipeng Li. Million-scale near-duplicate video retrieval system. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 837–838, 2011.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

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

[10] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[11] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[12] Matthijs Douze, Jérôme Revaud, Cordelia Schmid, and Hervé Jégou. Stable hyper-pooling and query expansion for event detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1825–1832, 2013.

[13] Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. Video re-localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018.

[14] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[15] Yanbin Hao, Tingting Mu, John Y Goulermas, Jianguo Jiang, Richang Hong, and Meng Wang. Unsupervised t-distributed video hashing and its deep hashing extension. *IEEE Transactions on Image Processing*, 26(11):5531–5544, 2017.

[16] Yanbin Hao, Tingting Mu, Richang Hong, Meng Wang, Ning An, and John Y Goulermas. Stochastic multiview hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*, 19(1):1–14, 2016.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[22] Yaocong Hu and Xiaobo Lu. Learning spatial-temporal features for video copy detection by the combination of cnn and rnn. *Journal of Visual Communication and Image Representation*, 55:21–29, 2018.

[23] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012.

[24] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[25] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. Vcdb: a large-scale database for partial copy detection in videos. In *European conference on computer vision*, pages 357–371. Springer, 2014.

[26] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501, 2007.

[27] Weizhen Jing, Xiushan Nie, Chaoran Cui, Xiaoming Xi, Gongping Yang, and Yilong Yin. Global-view hashing: harnessing global relations in near-duplicate video retrieval. *World wide web*, 22(2):771–789, 2019.

[28] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and

Dilip Krishnan. Supervised contrastive learning. *ArXiv*, abs/2004.11362, 2020.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Fivr: Fine-grained incident video retrieval. *IEEE Transactions on Multimedia*, 21(10):2638–2652, 2019.

[31] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6351–6360, 2019.

[32] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *International conference on multimedia modeling*, pages 251–263. Springer, 2017.

[33] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017.

[34] Yang Li, Yulong Xu, Jiabao Wang, Zhuang Miao, and Yafei Zhang. Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. *IEEE Signal Processing Letters*, 24(5):609–613, 2017.

[35] Kaiyang Liao, Hao Lei, Yuanlin Zheng, Guangfeng Lin, Congjun Cao, Mingzhu Zhang, and Jie Ding. Ir feature embedded bof indexing method for near-duplicate video retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(12):3743–3753, 2018.

[36] Hao Liu, Qingjie Zhao, Hao Wang, Peng Lv, and Yanming Chen. An image-based near-duplicate video retrieval and localization using improved edit distance. *Multimedia Tools and Applications*, 76(22):24435–24456, 2017.

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[38] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[42] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[43] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer, 2016.

[44] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

[45] Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Event retrieval in large video collections with circulant temporal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2459–2466, 2013.

[46] Omar Seddati, Stéphane Dupont, Saïd Mahmoudi, and Mahnaz Parian. Towards good practices for image retrieval based on cnn features. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1246–1255, 2017.

[47] Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *ArXiv*, abs/1802.05799, 2018.

[48] Lifeng Shang, Linjun Yang, Fei Wang, Kwok-Ping Chan, and Xian-Sheng Hua. Real-time large scale near-duplicate web video retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 531–540, 2010.

[49] Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, 2003.

[50] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[51] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 423–432, 2011.

[52] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Jiebo Luo. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*, 15(8):1997–2008, 2013.

[53] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020.

[54] Hung-Khoon Tan, Chong-Wah Ngo, Richard Hong, and Tat-Seng Chua. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 145–154, 2009.

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[56] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[58] Feng Wang. *Research on Deep Learning Based Face Verification*. PhD thesis, University of Electronic Science and Technology of China, 2018.

[59] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[60] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[61] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 218–227, 2007.

[62] Zhipeng Wu and Kiyoharu Aizawa. Self-similarity-based partial near-duplicate video retrieval and alignment. *International Journal of Multimedia Information Retrieval*, 3(1):1–14, 2014.

[63] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[64] Jin Xia, Jie Shao, Cewu Lu, and Changhu Wang. Weakly supervised em process for temporal localization within video. In *2019 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2019.

[65] Bingchen Zhao and Xin Wen. Distilling visual priors from self-supervised learning. In *Proceedings of the European Conference on Computer Vision Workshops*, 2020.

[66] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.

[67] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2017.

[68] Liang Zheng, Yali Zhao, Shengjin Wang, Jingdong Wang, and Qi Tian. Good practice in cnn feature transfer. *arXiv preprint arXiv:1604.00133*, 2016.