Learning Multi-scale Representations with Single-stream Network for Video Retrieval

Chia-Hui Wang[1], Yu-Chee Tseng[1,2], Ting-Hui Chiang[3], and Yan-Ann Chen[4]

[1]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[2]College of AI, National Yang Ming Chiao Tung University, Taiwan
[3]Advanced Technology Laboratory, Chunghwa Telecom Laboratories, Taoyuan, Taiwan
[4]Department of Computer Science and Engineering, Yuan Ze University, Taiwan
{qvqv621.cs08, yctseng}@cs.nycu.edu.tw, thchiang@cht.com.tw, chenya@saturn.yzu.edu.tw

Abstract

With the explosive growth of video contents in the Internet, video retrieval has become an important issue that can benefit video recommendation and copyright detection. Since the key features of a video may distribute in distant regions of a lengthy video, several works have made a success by exploiting multi-stream, multi-scale architectures to learn and merge distant features. However, a multi-stream network is costly in terms of memory and computing overhead. The number of scales and these scales are handcrafted and fixed once a model is finalized. Further, being more complicated, multi-stream networks are more prone to being overfitting and lead to poorer generalization. This paper proposes a single-stream network with built-in dilated spatial and temporal learning capability. By combining with modern techniques, including Denoising Autoencoder, Squeeze-and-Excitation Attention, and Triplet Comparative Mechanism, our model achieves state-of-the-art performance in several video retrieval tasks on the FIVR-200K, CC_WEB_VIDEO, and EVVE datasets.

1. Introduction

Various deep learning-based video applications have been developed [13, 41, 58, 59]. With the growth of user stickiness on video-sharing platforms, such as YouTube, Vimeo, TikTok, and Facebook, the amount of Internet video contents has increased rapidly on a daily basis. Compared to the other forms of media, videos are more interactive and entertaining. However, keeping track of videos is a challenge because of not only lack of labels, but also the way to assign proper labels. Video editing and forwarding further exacerbate the problem. The video retrieval task [1] is to identify the relevant video(s) of a given query from a video dataset. A query may be in a form of text [28, 62], audio [31, 62], image [2, 64], and/or video clip [12, 33, 34, 37].

In this work, we consider the video retrieval task where the query is also a video clip. In general, video retrieval involves three steps: feature extraction, feature aggregation, and similarity calculation [33]. The extraction of discriminative features is an essential step. Features can be obtained in a hand-crafted manner or from a learning-based approach. Handcrafted features can be local ones or global ones. Local approaches include local binary patterns (LBP) [32, 54], scale-invariant feature transform (SIFT) [29, 61], and speeded-up robust feature (SURF) [46]. Global approaches include Color Histograms [32,54,61], 3D-Discrete Cosine Transform [17], and TIRIs [20]. Learning-based approaches are proved to achieve higher performance recently. To identify spatiotemporal representations, models have been built based on CNN [27] and RNN [5, 21]. The work [30] uses pre-trained AlexNet [38] and Siamese CNN to extract global and local features. Reference [34] applies Regional Maximum Activation of Convolution (R-MAC) [35, 57] to find feature descriptors. In [12], an encoder-decoder ConvLSTM model that explores multi-embedding of a video is proposed. For feature aggregation, traditional approaches have combined frame-level features into video-level representations using Global Vector [19,23,35,40,61], hash codes [54, 56, 63] and Bag-of-Words [6, 35, 43]. However, these methods may be dominated by certain frames and disregard the temporal relation of frames. Other approaches consider frame transition information in similarity calculation, such as dynamic programming [16, 44], temporal networks [30], and temporal hough voting [18, 29].

We observe that extracting and merging distant features, both spatially and temporally, plays a critical role in video

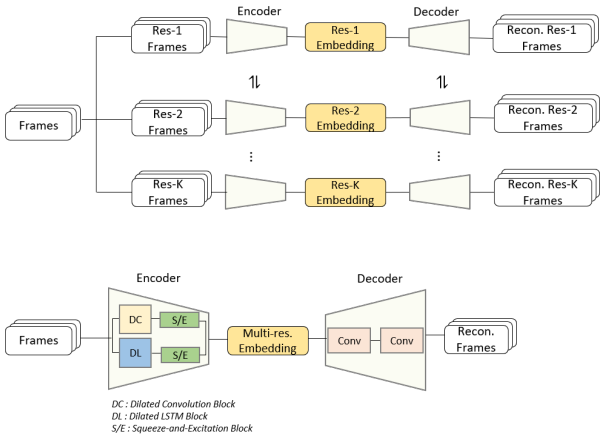


Figure 1. Basic idea: multi-stream (top) vs. single-stream architecture (bottom).

retrieval, especially for lengthy or edited videos. Most works resolve this issue by a multi-stream architecture, where each stream handles a particular spatial/temporal resolution [12]. Although effective, the number of resolutions and their resolutions are handcrafted and fixed once a model is finalized. In SSD (Single Shot MultiBox Detector) [45], a single-stream network for predicting the bounding boxes of objects at multiple scales is proposed. By applying dilated convolutions in certain network layers, the spatial receptive field is increased without reducing the resolution of feature maps and its multi-scale feature maps can handle different scales of objects. While being simpler and faster than multi-stream networks, SSD achieves remarkable performance on several benchmarks. The work [9, 10] uses dilated convolution or dilated recurrent skip connections as one of the key components in their architectures. They show that dilated operations can cover a larger receptive field without increasing the number of parameters and speed up the process of the network. By using different dilation rates, the network can extract features at different spatial and temporal scales.

Inspired by [9, 10, 45], we propose a video embedding network that tries to learn both dilated spatial and dilated temporal representations under a single-stream architecture. Fig. 1 illustrates the main idea. Our model is constructed by a denoising encoder-decoder framework, and the dilated convolution structure and the dilated LSTM structure capture multiscale fine- and coarse-grained spatiotemporal characteristics through learnable parameters. At the end, a comparative network trained with the triplet loss and binary cross-entropy loss calculates the similarity between a pair of videos.

2. Related Work

Generally, we obtain video representations by two processes, feature extraction and feature aggregation. Early work often builds video representations from the frame level. First, local feature-based approaches, such as SIFT [29, 61] and LBP [32, 54], generate frame-level descriptions. Second, global feature-based approaches, such as Color Histograms [61] and Auto Colour Correlograms [6], form representations from a video sequence containing both spatial and temporal information. Feature aggregation is to incorporate frame-level information into global representations. Popular aggregation methods include Global Vector [19, 23, 35, 40, 61], Fisher Vector [47, 50], and Bag-of-Words [6, 35, 43]. Global Vector can easily be controlled by frequently-appeared patterns as it simply averages all frames. Bag-of-Words is more discriminative since it creates the visual codebook by mapping each frame into visual words and utilizes the TF-IDF weighting scheme to acquire a video-level representation. Other approaches take into account the alignment of the temporal sequence using Temporal Hough voting [18, 29], Temporal Network [30], and Dynamic programming [16, 44]. These methods can perform well on regular patterns, but are not capable of capturing diverse patterns.

Recently, deep learning solutions have attracted more attention. The construction of video representations usually aggregates the results from pre-trained neural networks, e.g. Maximum Activation of Convolution (MAC) [34, 57] and variants of MAC [24, 35, 65]. To be rich in video representation, some approaches [21, 27] consider more spatiotemporal information by combining CNN with commonly used recurrent neural networks such as Long Short-Term Memory (LSTM) [25] and Gated Recurrent Unit (GRU) [14].

To recognize the similarities between video representations, distance metrics such as dot product and Euclidean distance can be applied. Several video hashing methods [55, 63] choose to use the Hamming distance. The performance of these methods depends on a suitable hash function. Recently, the approaches [34, 51] achieve competitive performance using Chamfer similarity between frame-level and video-level video descriptors. Other approaches such as the Bag-of-Words and its variants [6, 16] rank videos according to TF-IDF values. Contrastive learning learns from positive and negative pairs of data. Contrastive loss [15] is one of the earliest versions of loss functions used for deep metric learning (DML) [27, 36, 42]. The purpose is to minimize the embedding distances of the same class but to maximize those otherwise. Triplet loss, which involves one anchor, one positive, and one negative sample, encourages dissimilar pairs to be distant from similar pairs by at least a specific margin value. The recent training objective is to include multiple positive and negative pairs in one batch, such as Multi-Class N-pair loss [53] and Soft-Nearest Neighbors Loss [22, 49].

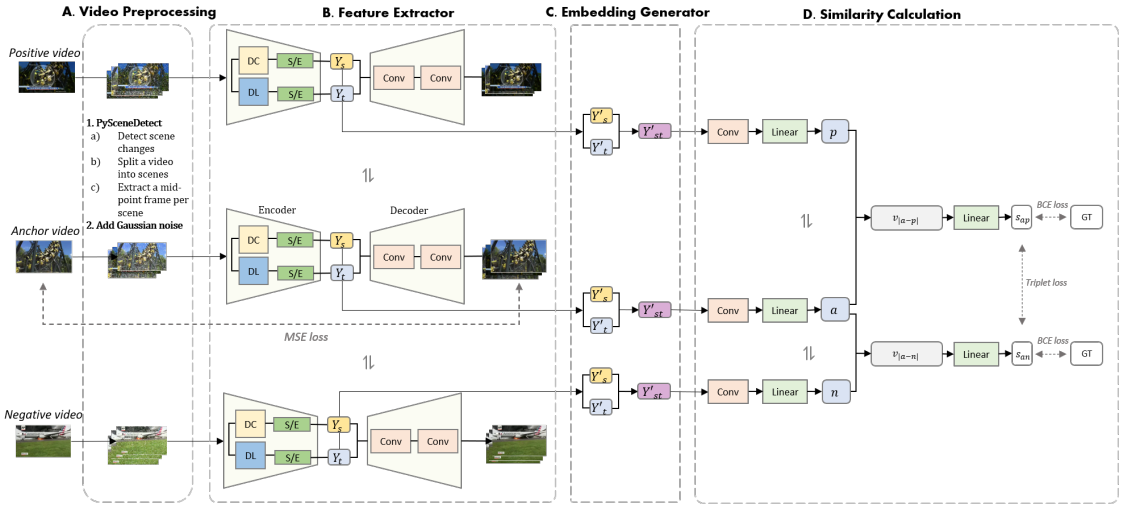


Figure 2. The proposed video retrieval architecture. The denoising autoencoder has a single-stream structure containing two dilated feature extraction paths.

3. Proposed Architecture

Given a query video, the objective is to rank all videos in the database according to their relevance to the query. Fig. 2 shows the proposed architecture. The model consists of four components: video preprocessing, feature extractor, embedding generator, and similarity calculation. The video preprocessing samples representative frames. The feature extractor is a single stream network with a denoising autoencoder as its backbone. It prevents learning an identity mapping as opposed to a standard autoencoder and still performs well on corrupted data. There are three loss functions in our model that include Mean Squared Error (MSE) to compare the original frame sequences and the recovered output from the decoder, and triplet loss combined with binary cross entropy for the similarity score calculation step. Below, we introduce the details. Throughout our presentation, the notations used are as follows: X (input tensor), Y (output tensor), N (batch size), T (number of frames), H (height), W (width), and C (channels).

3.1. Video Preprocessing

Videos may have different lengths. For each video, we select representative frames by partitioning it according to scene changes. To do so, there are several scene detection methods available in the open source tool PySceneDetect [7,8]. We choose *content-aware scene detector*, which identifies a scene change based on the frame-to-frame difference in edges and HSV colors (hue, saturation, and brightness). If the difference exceeds a preset threshold, a scene change is detected. Then, from each scene, we select the midpoint frame as its representative. The purpose is to retain meaningful information per scene. These midpoint frames form the trimmed video. As frames of videos may be of different sizes, we resize each frame to the lowest height and width in the dataset. Then we add Gaussian noise to these frames. The frame sequence after the above processing, denoted by a 4-D (T, H, W, C) tensor per video, is the actual input to the model.

3.2. Feature Extractor

The goal of the feature extractor is to retrieve representative features from a video. Fig. 3 shows the extractor's architecture. It has an encoder and a decoder. The encoder starts with two common convolution layers. This is because the subsequent dilation part is relatively wide and shallow; increasing the depth can help learn more intermediate features, leading to better generalization capability. It follows by two excitation paths, both with a dilation design, for identifying rich and distant features among frames. At the ends of both paths, a squeeze-and-excite module is added to improve the attention capability of our model. The excitation results are denoted by Y_s and Y_t, respectively. Finally, the decoder is designed to recover noisy frames back to noise-free original frames by $Y_s|Y_t$. Below, we detail the two dilation paths.

3.2.1 Dilated Convolutional Excitation Path

Dilated convolution blocks have been explored by [3, 10, 11, 39], with the aim of expanding the field of view of filters without increasing the computation or the number of parameters of a model. Unlike previous work designed for 1-D time series and 2-D images, we apply the structure to our 3-D video tensors. Then an attention mechanism follows.

To emphasize the diversity in spatial resolution, we use a hierarchy of 3D dilated convolution filters (sliding along

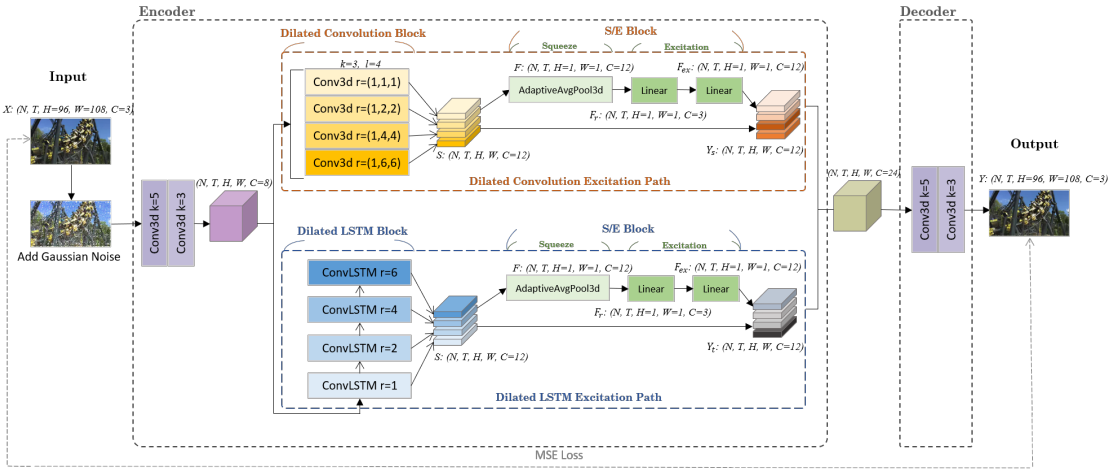


Figure 3. The architecture of denoising autoencoder. It has a dilated convolutional excitation path and a dilated LSTM excitation path.

T, H, and W) instead of pooling or downsampling mechanisms. The standard convolution has a dilation rate $r = 1$. In our case, we explore spatial dimensions (H and W) and fix the temporal dimension (T). Thus, we set our dilation rate as $r = (1, H, W)$. Given an input $X \in \mathbb{R}^{N\times T\times H\times W\times C}$, we apply parallel convolution layers with different dilation rates to X and then stack the results with respect to the channel axis. Dilated convolutions expand the receptive field without loss of resolution at the output layer (we do not choose to use pooling and stride convolutions because they would reduce resolution). We set the padding the same as the dilation rate to fix dimensions H and W. There are l parallel layers, each with k kernels. So the final stacked tensor $S \in \mathbb{R}^{N\times T\times H\times W\times C'}$, which $C' = k \times l$.

The above parallel dilated convolutions enrich S, but also add lots of channels to S ($k \times l$). Therefore, applying weights to C' is meaningful. Inspired by SENet [26], we adopt the Squeeze-and-Excitation (SE) network for channel attention. So tensor S is fed into a SE block to learn the importance levels of channels. In the *squeeze* operation, we opt for a global average pooling to generate channel-wise statistics:

$$F = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} S[:, :, i, j, :] \tag{1}$$

where $F \in \mathbb{R}^{N\times T\times 1\times 1\times C'}$. In the *excitation* operation, we first reduce the number of channels of F by a ratio r to $F_r \in \mathbb{R}^{N\times T\times 1\times 1\times \frac{C'}{r}}$. To map the scaling weights and project the output back to the same dimension as S, we unsqueeze F_r by employing fully connected layers with Sigmoid activation:

$$F_{ex} = \sigma(W_2 \delta(W_1 F)) \tag{2}$$

where σ refers to the Sigmoid function, δ refers to the ReLU function, $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, and $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$. Subsequently,

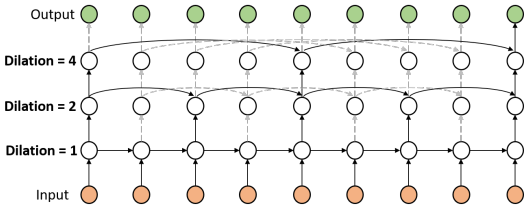


Figure 4. An illustration of dilated connections of a multi-layer RNN.

the output F_{ex} is applied to S by simple element-wise multiplication. The final weighted tensor is denoted as Y_s.

3.2.2 Dilated LSTM Excitation Path

The design of dilation is similar to the previous path, except that we shall focus on the temporal axis (T). This dilated block is derived based on [9, 52] and we stack ConvLSTM layers with dilations. ConvLSTM is a type of recurrent neural network for spatiotemporal prediction. It applies convolution operations instead of matrix multiplications. The key equations of ConvLSTM are shown below, where H_t is the hidden state, C_t is the cell state, X_t is input information, i_t, f_t, and o_t are various gates, $*$ denotes the convolution operator, and $\odot$ denotes the Hadamard product:

$$\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
H_t &= o_t \odot \tanh(C_t)
\end{aligned} \tag{3}$$

We adopt the dilated recurrent skip connection design proposed in [9] to extract nearby as well as distant temporal data dependencies. The basic idea is illustrated in Fig. 4.

We base our structure upon the stack and temporally concatenate multiple ConvLSTM units, in which the output of a ConvLSTM hidden layer is fed as the input into the subsequent ConvLSTM hidden layer. To capture distant temporal features, we skip a different number of timestamps after each layer hierarchy. This forms a pyramid-like structure such that every layer can focus on different temporal resolutions. The first and the bottom layers build the solid and overall information base, which are the same as the vanilla ConvLSTM cells, having a dilation of $r = 1$. For a higher hidden layer l, we set its dilation as $r = (l - 1) \times 2$, which means fetching the previous state output by skipping $r - 1$ timestamps. For example, in Fig. 4, the dilation rates are $r = 1, 2,$ and 4. The deep architecture allows quickly fetching distant temporal features from a frame sequence. Since we intend to have the final output contain a variety of receptive fields, we concatenate the outputs from all the layers to form the final tensor S. This design provides multiscale receptive fields without changing the sizes of kernels/filters, helping our model memorize historical information and tackle the challenge of vanishing and exploding gradients.

The rest of the module also contains a squeeze-and-excitation network. The steps are similar to the previous path. It contains applying S to find tensors F, F_r, and F_{ex} for attention purpose. So we omit the details. Finally, F_{ex} is applied to S by element-wise multiplications. The final weighted tensor is denoted by Y_t.

3.3. Embedding Generator

After the denoising autoencoder has been well trained, we take out the trained encoder and use it to generate the embedding of a video. Rather than extracting encoding output directly, we translate it from a high-dimensional space to a low-dimensional one by taking the average among the channel and temporal axes, in hope of keeping semantic meanings. Recall that the outputs of the spatial and the temporal excitation paths are Y_s and Y_t, respectively. To obtain the embedding, we reshape Y_s and Y_t as follows. First, we compute the mean across the dimensions *T* and *C* and reshape the tensor *(N, T, H, W, C)* to *(N, H, W)*, denoted as Y_s' and Y_t'. Now the two tensors have the same size. Then, we concatenate them, resulting in the embedding $Y_{st}' \in (N, 2, H, W)$. Note that the second axis (2) is regarded as the channel axis, which will facilitate the processing of the 2D convolution layer in the upcoming similarity calculation.

3.4. Similarity Calculation

Similarity calculation is based on a triplet comparative model with flows of anchor, positive, and negative videos. The embedding obtained above is able to distinguish, as well as recovery, a video. However, our purpose is to use it to compare the similarity of two videos. Therefore, we transform it with a 2D convolution layer and two fully connected layers. These layers are to learn the mappings from video embedding to a compact vector that is able to measure the distance, i.e., similarity, between videos. The input tensor from the previous phase is *(N, 2, H, W)*. The convolution layer first reduces the tensor's height, width, and channel by a factor of 2. Then flatten it and connect to the fully connected layers to generate a 1-dimensional vector with 1024 nodes. Note that the triplet scheme has only one model, whose weights are shared by three input tensors: anchor, positive, and negative. Our distance vectors are obtained by calculating the absolute value of the element-wise subtraction, which can be indicated as follows: the anchor-positive and anchor-negative distances are $v_{|a-p|}$ and $v_{|a-n|}$, respectively. Then we feed these distance vectors to the last fully connected layer to generate one node to compute a similarity score s_{ap} and s_{an}.

3.5. Loss Functions

There are three loss functions associated with our model, as reflected in Fig. 2. We choose MSE for the optimization of the denoising autoencoder, which is the mean of the squared difference between a denoised video and its ground truth:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

where n is the number of frames, y_i is the i-th frame of the ground truth sequence and $\hat{y}_i$ is the recovered frame.

To optimize the comparative model, we leverage both the triplet loss and the binary cross-entropy loss. Traditional triplet loss is applied on the distance vector, which means minimizing the distance from the anchor to the positive and maximizing the distance from the anchor to the negative with a margin threshold. In our case, we force the network to directly result in higher similarity scores for the similar video pairs and lower for the dissimilar ones:

$$L_{tr} = max(s_{an} - s_{ap} + m, 0) \tag{5}$$

where m is a margin value. Then we also use the binary cross-entropy to compare each of the predicted probabilities to actual output. The formula is the negative average of the log of corrected predicted probabilities:

$$BCE = -[g \log x + (1 - g) \log(1 - x)] \tag{6}$$

where g is the ground truth, and x is the predicted value ranging between 0 and 1. We set the same weight to L_{tr} and BCE.

4. Experiment Results

The proposed approach is evaluated on three datasets and compared with several state-of-the-art methods. In all cases, we use the mean Average Precision (mAP) metric.

4.1. Datasets

FIVR-200K consists of 225,960 videos associated with 4,687 Wikipedia events and 100 queries, which has been used as a benchmark dataset for fine-grained incident video retrieval. In this dataset, they define three types of retrieval tasks: (1) Duplicate Scene Video Retrieval (DSVR): Two videos that are regarded as the "same incident" share at least one scene captured from the same camera but regardless of any applied transformation. Relevant videos are annotated with ND (near-duplicate) and DS (duplicate-scene) labels. (2) Complementary Scene Video Retrieval (CSVR): Two videos that are regarded as the "same incident" contain at least one spatiotemporal overlapping segment and can be captured from different viewpoints. The task considers the labels ND, DS, and CS (complementary-scene) as relevant. (3) Incident Scene Video Retrieval (ISVR): Two videos that are regarded as the "same incident" can be spatially and temporally close but have no overlap, where ND, DS, CS, and IS (incident-scene) are considered relevant. In conclusion, the relationship between three tasks is DSVR $\subseteq$ CSVR $\subseteq$ ISVR and we assume the difficulty as follows DSVR $\leq$ CSVR $\leq$ ISVR. We managed to download 63,247 videos and 36 queries due to some videos are missing and deleted.

CC_WEB_VIDEO contains 24 query sets and 12,790 videos. This dataset focuses on duplicate and near-duplicate videos from the Web video search, which are approximately identical videos with some image transformations and modifications. Each query video is chosen by the most popular video in the set of clips. In this dataset, they detect the boundaries of the shot and provide keyframes of each shot. The study [34] provides a cleaned version of the dataset since the original one has some mislabeled ones in the annotations. Our model evaluates both the original version and the cleaned version.

EVVE was developed for the problem of event video retrieval, which consists of 2,375 Youtube videos and 620 queries. The dataset contains 13 major events, such as human and natural events, which were annotated by the first annotator that produced the precise definition. Due to the reason for the removal of the video from YouTube, only 1,732 videos and 377 queries were downloaded.

4.2. Implementation Details

For frame extraction, we download the keyframes provided in the CC_WEB_VIDEO dataset and use the PySceneDetect tool to detect scene changes for the other two datasets. Then we resize frames according to the lowest resolution in the dataset. The frame size $H \times W$ is 96×108 for FIVR-200K and CC_WEB_VIDEO, and 112×160 for EVVE. For the encoder-decoder model, we build our dilated convolution and dilated LSTM blocks with kernel size 3 and implement 4 stacked layers with the dilation rates of $(1, 2, 4, 6)$. The reduction rate in the following S/E block is set as 4 for our baseline model. For the decoder, we build two simple convolution layers. During the training process, we adopt the Adam optimizer for our denoising autoencoder and comparative model, with the learning rate set to 10^{-2}.

Since the number of representative frames per video may vary (according to the scene detection results), the input shape of each video may also vary. Although the encoder is able to process videos of variable lengths, the batch size of each iteration may differ from each other. To solve the dynamic input shape problem, we design a batch generator that only includes video clips of the same length in the same batch and that limits the maximum number of frames per batch to a constant 1000. Then we fill in equal-length video clips into a batch until the limit 1000 is reached. Note that while the lengths of the embeddings of videos may differ, the embedding generator will compress the dimension T, making tensors of the same size for comparison. To train the comparative model, we set the batch size to 128.

We train the denoising autoencoder for about 20-25 epochs, and the comparative model for about 10-15 epochs. Using the Early-Stopping callback function, we reduce the training time and save the best model with the lowest validation loss for further evaluation. There is a serious imbalanced issue in these datasets because irrelevant video pairs are much more than relevant pairs, making model training more difficult. For the FIVR-200K dataset, we randomly pick some 0-labelled video pairs to equal the number of replicated 1-labelled video pairs. We exploit MSE loss function for denoising autoencoder and triplet loss with BCE loss function for similarity calculation. The margin value we set in the triplet loss function is 0.9.

To evaluate the quality of video rankings, we use the mean Average Precision (mAP) metric as defined in [33]. The Average Precision (AP) for each query is calculated as

$$AP = \frac{1}{N} \sum_{i=0}^{N} \frac{i}{r_i} \tag{7}$$

where N is the number of relevant videos to the query video and r_i is the rank of the *i*-th retrieved relevant video. The mAP is calculated by averaging the AP scores of all queries in the dataset. For example, $AP = 1$ means that all N relevant videos are ranked in the first N of the list. The higher the AP score, the higher the retrieval accuracy.

4.3. Comparisons with State-of-the-Arts

The proposed approach is compared with several state-of-the-art methods on FIVR-200K, CC_WEB_VIDEO, and EVVE. The comparison results are given in Tab. 1, Tab. 2, and Tab. 5, respectively.

FIVR-200K: We compare our model against the following state-of-the-arts. Deep Metric Learning (DML)

Method	DSVR	CSVR	ISVR
DML [36]	0.3460	0.3293	0.2880
LBoW [35]	0.6123	0.5858	0.5208
UTS+FRP [42]	0.7686	0.7239	0.6127
ViSiL [34]	0.8790	0.8475	0.7210
A-DML [60]	0.627	-	-
TCA [51]	0.877	0.830	0.703
DnS [37]	0.921	0.875	0.741
Multi-2 [12]	0.8819	0.8795	0.7898
Ours	**0.9240**	**0.8807**	**0.8356**

Table 1. mAP comparisons of three video retrieval tasks on FIVR-200K.

Method	CC_WEB	CC_WEB_cleaned
PPT [16]	0.958	-
CTE [48]	**0.996**	-
DML [36]	0.971	0.979
ViSiL [34]	0.985	**0.996**
DnS [37]	0.984	0.995
TCA [51]	0.983	0.994
Multi-2 [12]	0.976	0.986
Ours	0.975	0.986

Table 2. mAP comparisons of the NDVR task on CC_WEB_VIDEO.

Task	DSVR	CSVR	ISVR
1P	651,381	654,862	726,809
2P	1,283,283	1,278,868	1,419,085
4P	2,556,416	2,531,028	2,800,476

Table 3. Ablation study on data replication: numbers of triplets after replication (FIVR-200K).

[36] trains a network using the triplet loss scheme; Layer Bag-of-Words (LBoW) [35] compacts visual information based on BoW schemes incoporating with tf-idf weighting; UTS+FRP [42] proposes an unsupervised teacher-student model and a frame-level retrieval pipeline to acquire discriminative video representations; ViSiL [34] contributes a similarity computation method by combining a frame-to-frame scheme with video-to-video scheme; A-DML [60] proposes a two-stream attention network for RGB and combines optical flow features based on DML approach; TCA [51] applies contrastive learning to train a transformer-based architecture; DnS [37] trains several student networks via a Teacher-Student setup at performance-efficiency trade-offs; Muiti-2 [12] develops a multi-stream encoder-decoder ConvLSTM model to extract spatiotemporal embeddings. As can be seen, the proposed model is superior to the state-of-the-arts in all three tasks.

CC_WEB_VIDEO: This dataset simulates the NDVR problem. We provide the comparison results on both the original and the cleaned versions. We compare to DML, ViSiL, DnS, TCA, and Multi-2. Two more approaches are included in the comparison: PPT [16] presents a re-ranking pattern-based method with a BoW-based scheme; CTE [48] encodes spatiotemporal representations by the Fourier transform. Tab. 2 shows that our model remains quite competitive compared to these state-of-the-arts.

Task	DSVR	CSVR	ISVR
1P	0.6083	0.5424	0.3502
2P	0.7802	0.7016	0.6870
4P	**0.9240**	**0.8807**	**0.8356**

Table 4. Ablation study on data replication: comparison on mAP by varying replication times (FIVR-200K).

EVVE: We calculate the mAP per event since in this dataset each event has several query videos. The order of the event categories is the same as in [48]. The difficult part of this data set is that each event has particular depictions and characteristics. For example, #2 describes the wedding of Prince William and Kate Middleton, which counts a slideshow in other weddings as positive; #10 describes the major autumn flood in Thailand in 2011, which covered video of the flood in different places; and #12 describes the eruption of the Strokkur geyser in Iceland, which are reoccurring events. The comparisons in Tab. 5 show that our method performs better in events #3, #6-#10, and the average score, indicating the ability of our model to integrate distant visual information and semantic meanings.

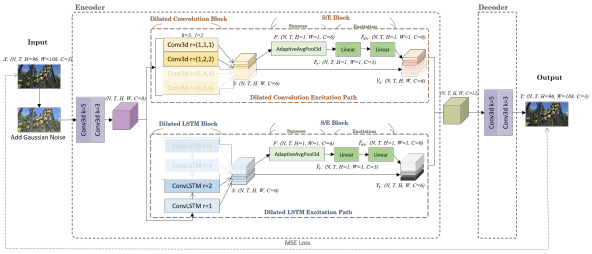
We measure the feature extractor model size and FLOPs of our and Multi-2 to get more insight into single- and multi-stream networks. The number of parameters of one single-resolution stream in Multi-2 is 15,371 and the FLOPs is 0.632G. Our model with complete two paths (i.e., four spatial resolutions and four temporal resolutions) has 34,746 parameters and 3.59G FLOPs.

4.4. Ablation Experiments

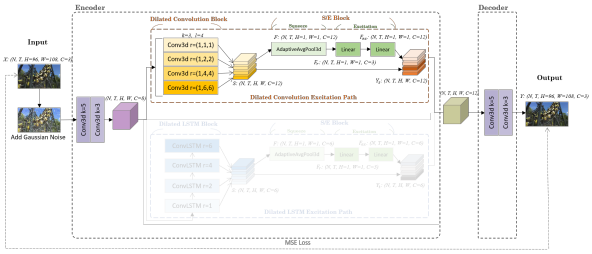
There is an imbalanced data issue in these datasets–a video usually has much less relevant videos than irrelevant ones. To alleviate the insufficient positive pairs problem, we replicate 1-labelled video pairs 2 or 4 times to form a dataset. We conduct an experiment on FIVR-200K. Among all the triplets, each positive pair can contribute 1, 2, or 4 times to the dataset (the negative pair of a triplet is selected in a random manner since they outnumber the former). Tab. 3 shows the numbers of triplets in our three datasets after such replications, where P is the original number of positive pairs and 2P and 4P mean the two replication cases. Tab. 4 shows the effectiveness of such replications–we achieve the best performance in the case of 4P. Although using more replications may gain further improvement, we

Method	avg	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13
CTE [48]	0.376	0.694	0.394	0.111	0.486	0.260	0.281	0.202	0.132	0.212	0.371	0.246	0.774	0.719
LAMV [4]	0.587	0.837	0.500	0.126	0.588	0.455	0.343	0.267	0.142	0.230	0.293	0.216	**0.950**	0.776
TCA [51]	0.630	-	-	-	-	-	-	-	-	-	-	-	-	-
ViSiL [34]	0.631	**0.918**	**0.724**	0.227	0.446	0.390	0.405	0.308	0.223	0.604	0.578	0.399	0.916	**0.855**
DnS [37]	0.651	-	-	-	-	-	-	-	-	-	-	-	-	-
Multi-2 [12]	0.6561	0.5001	0.6027	0.2993	**0.7322**	**0.6828**	0.7045	0.7957	0.5299	0.5312	0.6072	**0.6824**	0.6563	0.6039
Ours	**0.6852**	0.8137	0.6576	**0.6910**	0.6730	0.6713	**0.7193**	**0.8471**	**0.6591**	**0.7013**	**0.8802**	0.6776	0.6788	0.6858

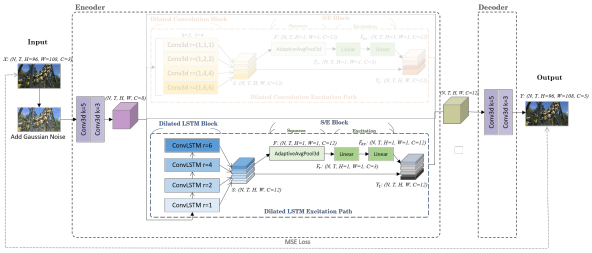
Table 5. mAP comparisons on EVVE. (**avg** means the average of #1 $\sim$ #13)



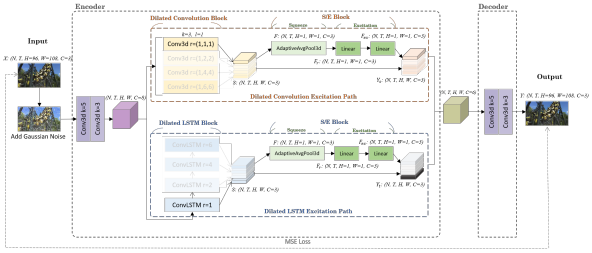
(a) DCDL2 with two dilation rates of (1,2)



(b) DC with upper path only



(c) DL with lower path only



(d) CL with no dilation

Figure 5. Ablative study cases. Shaded areas are not implemented.

Method	DSVR	CSVR	ISVR
DCDL4	**0.9240**	**0.8807**	**0.8356**
DCDL2	0.8694	0.8333	0.7762
DC	0.8162	0.8053	0.7877
DL	0.7829	0.7758	0.7640
CL	0.7924	0.7748	0.6797

Table 6. Ablation study on the model architecture (FIVR-200K).

found 4P to achieve an excellent trade-off among performance, computing speed, and storage requirement. Therefore, we choose 4P in our experiments.

Next, we investigate the contributions of different modules in our model. We name the model in Fig. 2 DCDL4 since there are four stacked Conv3d layers and four stacked ConvLSTM layers in the upper and the lower paths, respectively (i.e., four dilation rates). We test the four configurations in Fig. 5 on FIVR-200K. DCDL2 keeps the two-path structure, but has only two dilation rates (1 and 2) in each path. DC keeps the whole dilated convolutional excitation path, but deletes the lower path. DL keeps the whole dilated LSTM excitation path, but deletes the upper path. To validate the necessity of multiple resolutions, CL keeps the two-path structure, but there is no dilation at all. Tab. 6 presents the comparisons. DCDL4 performs the best. CL performs the worst, which validates the importance of using dilation. DC outperforms DL, which indicates that spatial characteristics play a more important role than temporal characteristics in the video retrieval task. DCDL2 surpasses DC and DL, which indicates the notable impact of using our two-path excitation structure.

5. Conclusions

In this paper, we have proposed a denoising autoencoder framework for the video retrieval problem that employs horizontally stacked dilated convolution and dilated LSTM layers with the attention mechanisms. The model learns the multi-scale spatiotemporal representation of a video in a single-stream network. The model outperforms the other non-dilated counterparts that employ a multi-stream, multi-resolution approach in most retrieval sub-tasks except the NDVR sub-task. Recently, the transformer-based structure is also proved to be able to handle distant features well, which deserves further study.

Acknowledgement: This work was sponsored by NSTC grant 111-2634-F-A49 -013, Taiwan.

References

[1] Aasif Ansari and Muzammil H Mohammed. Content based video retrieval systems-methods, techniques, trends and challenges. *International Journal of Computer Applications*, 112(7), 2015. 1

[2] Andre Araujo and Bernd Girod. Large-scale video retrieval using image queries. *IEEE transactions on circuits and systems for video technology*, 28(6):1406–1420, 2017. 1

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 3

[4] Lorenzo Baraldi, Matthijs Douze, Rita Cucchiara, and Hervé Jégou. LAMV: Learning to align and match videos with kernelized temporal layers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7804–7813, 2018. 8

[5] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*, 2019. 1

[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. 1, 2

[7] Brandon Castellano. PySceneDetect. `http://scenedetect.com/en/latest/`, 2014. [Free and Open-Source Software; licensed under BSD 3-Clause]. 3

[8] Brandon Castellano. PySceneDetect. `https://scenedetect.com/en/latest/reference/detection-methods/`, 2014. [Free and Open-Source Software; licensed under BSD 3-Clause]. 3

[9] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *Advances in neural information processing systems*, 30, 2017. 2, 4

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2, 3

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 3

[12] Ting-Hui Chiang, Yi-Chun Tseng, and Yu-Chee Tseng. A multi-embedding neural model for incident video retrieval. *Pattern Recognition*, page 108807, 2022. 1, 2, 7, 8

[13] Sheng-Yang Chiu, Yu-Ting Huang, Chieh-Ting Lin, Yu-Chee Tseng, Jen-Jee Chen, Meng-Hsuan Tu, Bo-Chen Tung, and YuJou Nieh. Privacy-preserving video conferencing via thermal-generative images. In *IEEE International Conf. on Robotics and Automation (ICRA)*, 2023. 1

[14] 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. 2

[15] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 2

[16] 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. 1, 2, 7

[17] Baris Coskun, Bulent Sankur, and Nasir Memon. Spatio–temporal transform based video hashing. *ieee Transactions on Multimedia*, 8(6):1190–1208, 2006. 1

[18] Matthijs Douze, Hervé Jégou, and Cordelia Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Transactions on Multimedia*, 12(4):257–266, 2010. 1, 2

[19] Matthijs Douze, Hervé Jégou, Cordelia Schmid, and Patrick Pérez. Compact video description for copy detection with precise temporal alignment. In *European Conference on Computer Vision*, pages 522–535. Springer, 2010. 1, 2

[20] Mani Malek Esmaeili, Mehrdad Fatourechi, and Rabab Kreidieh Ward. A robust and fast video copy detection system using content-based fingerprinting. *IEEE Transactions on information forensics and security*, 6(1):213–226, 2010. 1

[21] 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. 1, 2

[22] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pages 2012–2020. PMLR, 2019. 2

[23] Zhanning Gao, Gang Hua, Dongqing Zhang, Nebojsa Jojic, Le Wang, Jianru Xue, and Nanning Zheng. Er3: A unified framework for event retrieval, recognition and recounting. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2253–2262, 2017. 1, 2

[24] 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. 2

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[27] 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. 1, 2

[28] CV Jawahar, Balakrishna Chennupati, Balamanohar Paluri, and Nataraj Jammalamadaka. Video retrieval based on textual queries. In *Proceedings of the thirteenth international*

conference on advanced computing and communications, Coimbatore, 2005. 1

[29] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. Vcdb: a large-scale database for partial copy detection in videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 357–371. Springer, 2014. 1, 2

[30] Yu-Gang Jiang and Jiajun Wang. Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE Transactions on Big Data*, 2(1):32–42, 2016. 1, 2

[31] Qin Jin, Peter Schulam, Shourabh Rawat, Susanne Burger, Duo Ding, and Florian Metze. Event-based video retrieval using audio. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012. 1

[32] 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*, 22:771–789, 2019. 1, 2

[33] 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. 1, 6

[34] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6360, 2019. 1, 2, 6, 7, 8

[35] 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. 1, 2, 7

[36] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 347–356, 2017. 2, 7

[37] Giorgos Kordopatis-Zilos, Christos Tzelepis, Symeon Papadopoulos, Ioannis Kompatsiaris, and Ioannis Patras. Dns: Distill-and-select for efficient and accurate video indexing and retrieval. *International Journal of Computer Vision*, 130(10):2385–2407, 2022. 1, 7, 8

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[39] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017. 3

[40] Hyodong Lee, Joonseok Lee, Joe Yue-Hei Ng, and Paul Natsev. Large scale video representation learning via relational graph clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6807–6816, 2020. 1, 2

[41] Jia-Yan Li, Jaden Chao-Ho Lin, Kun-Ru Wu, and Yu-Chee Tseng. Sensepred: Guiding video prediction by wearable sensors. *IEEE Internet of Things Journal*, 10(6):4698–4707, 2023. 1

[42] Dong Liang, Lanfen Lin, Rui Wang, Jie Shao, Changhu Wang, and Yei-Wei Chen. Unsupervised teacher-student model for large-scale video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 7

[43] 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. 1, 2

[44] 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*, 76:24435–24456, 2017. 1, 2

[45] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[46] Xiushan Nie, Weizhen Jing, Lin Yuan Ma, Chaoran Cui, and Yilong Yin. Two-layer video fingerprinting strategy for near-duplicate video detection. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 555–560. IEEE, 2017. 1

[47] Florent Perronnin and Diane Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3743–3752, 2015. 2

[48] 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. 7, 8

[49] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419. PMLR, 2007. 2

[50] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013. 2

[51] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. Temporal context aggregation for video retrieval with contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3268–3278, 2021. 2, 7, 8

[52] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 4

[53] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016. 2

[54] 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. 1, 2

[55] 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. 2

[56] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7):3210–3221, 2018. 1

[57] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 1, 2

[58] Yu-Yun Tseng, Po-Min Hsu, Jen-Jee Chen, and Yu-Chee Tseng. Computer vision-assisted instant alerts in 5g. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9, 2020. 1

[59] Lan-Da Van, Ling-Yan Zhang, Chun-Hao Chang, Kit-Lun Tong, Kun-Ru Wu, and Yu-Chee Tseng. Things in the air: Tagging wearable iot information on drone videos. *Discover Internet of Things*, 1:1–13, 2021. 1

[60] Kuan-Hsun Wang, Chia-Chun Cheng, Yi-Ling Chen, Yale Song, and Shang-Hong Lai. Attention-based deep metric learning for near-duplicate video retrieval. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5360–5367. IEEE, 2021. 7

[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. 1, 2

[62] Haojin Yang and Christoph Meinel. Content based lecture video retrieval using speech and video text information. *IEEE transactions on learning technologies*, 7(2):142–154, 2014. 1

[63] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3083–3092, 2020. 1, 2

[64] Chengyuan Zhang, Yunwu Lin, Lei Zhu, Anfeng Liu, Zuping Zhang, and Fang Huang. Cnn-vwii: An efficient approach for large-scale video retrieval by image queries. *Pattern Recognition Letters*, 123:82–88, 2019. 1

[65] 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. 2