Semi-supervised 3D Video Information Retrieval with Deep Neural Network and Bi-directional Dynamic-time Warping Algorithm

Yintai Ma [1] Diego Klabjan [1]

Abstract

This paper presents a novel semi-supervised deep learning algorithm for retrieving similar 2D and 3D videos based on visual content. The proposed approach combines the power of deep convolutional and recurrent neural networks with dynamic time warping as a similarity measure. The proposed algorithm is designed to handle large video datasets and retrieve the most related videos to a given inquiry video clip based on its graphical frames and contents. We split both the candidate and the inquiry videos into a sequence of clips and convert each clip to a representation vector using an autoencoder-backed deep neural network. We then calculate a similarity measure between the sequences of embedding vectors using a bi-directional dynamic time-warping method. This approach is tested on multiple public datasets, including CC_WEB_VIDEO, Youtube-8m, S3DIS, and Synthia, and showed good results compared to state-of-the-art. The algorithm effectively solves video retrieval tasks and outperforms the benchmarked state-of-the-art deep learning model.

Keywords: 3D Video Information Retrieval, Video Similarity Search, Unsupervised Learning, Semi-supervised Learning, Convolutional and Recurrent Neural Networks, End-to-end autoencoder

1. Introduction

In recent years, the exponential growth in online video data has made efficient retrieval of visually similar videos increasingly challenging. To address this, we propose a novel semi-supervised deep learning framework for retrieving similar 2D and 3D videos.

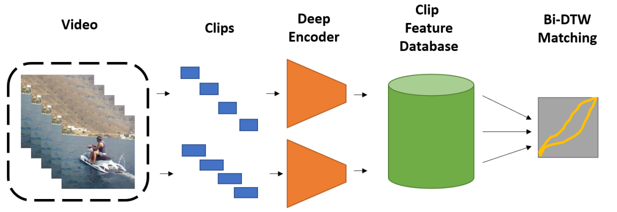


Figure 1. The proposed video representation learning pipeline first splits videos into short clips at a fixed number of frames. Then the proposed method converts clips to embedded feature vectors by a deep neural network encoder. Finally, we store all feature vectors in a database and use a bi-directional dynamic time-warping method to retrieve a list of candidates.

Our video retrieval approach represents videos as sequences of embedding vectors generated by a deep neural network encoder. It begins by splitting both candidate and query videos into fixed-length consecutive clips. Each clip is fed into the encoder to produce a representation vector. The sequence of embedding vectors for the full video is then compared to candidate videos using a bidirectional dynamic time warping similarity measure. By transforming videos into informative embedding sequences, and leveraging deep neural networks with dynamic time warping, our framework can effectively retrieve visually similar 2D and 3D videos.

A video embedding is created through the use of deep convolutional and recurrent neural networks, which are designed to extract rich and discriminative features from video clips. To optimize the video retrieval performance, we adopt a two-stage training approach. First, we pre-train the model unsupervised on a larger and relevant video dataset. Second, we fine-tune the model with a triplet loss function in a supervised manner, further enhancing its ability to perform video retrieval.

Finally, we compute the similarity between the sequences of embedding vectors between the query video and candidate videos using a variant of the dynamic time warping method. The bi-directional Dynamic Time Warping (Bi-DTW) method has been employed as a means to address the

[1] Department of Industrial Engineering and Management Science, Northwestern University, Evanston, United States. Correspondence to: Yintai Ma <yintaima2022@u.northwestern.edu>, Diego Klabjan <d-klabjan@northwestern.edu>.

limitations of the standard Dynamic Time Warping (DTW) in video embedding matching. The standard DTW algorithm, although effective at time series alignment, operates only in a single, forward direction. This unidirectional characteristic can potentially limit the accuracy of its matching results, particularly in applications such as video retrieval where the temporal structure of the data is complex and multidimensional. To mitigate these limitations, Bi-DTW was developed, with a distinguishing feature being its capacity to facilitate matching from both forward and backward directions. The rationale for this approach is rooted in the intuition that different portions of a video may align more effectively when approached from various temporal perspectives. Therefore, by integrating both forward and backward alignments, Bi-DTW improves the accuracy of video retrieval by ensuring a more robust, comprehensive temporal match. It presents a significant improvement over the conventional DTW, especially in complex, time-structured applications like video retrieval where the objective is to maximize the accuracy of the matching results. This allows us to determine the degree of similarity between the two videos, and effectively rank the candidate videos based on their relevance to the query video.

We evaluate the proposed method on several publicly available datasets, including CC_WEB_VIDEO, Youtube-8m, S3DIS, and Synthia, and show good results in comparison to state-of-the-art.

1.1. Contributions

We present a cutting-edge approach to video information retrieval by introducing a bi-directional dynamic time-warping method for determining the similarity between video inquiries. This innovative technique effectively tackles the temporal dimension of videos, resulting in improved accuracy and efficiency of the retrieval process.

Furthermore, this research encompasses the handling of 3D video inquiry, introducing both a novel 3D network architecture that expands upon 2D video information retrieval models and a method to incorporate 3D video data as an additional depth layer. This comprehensive framework offers a more robust solution for retrieving 3D video data.

We also introduce a sample retraining method that effectively addresses the challenge of handling difficult video pairs during the training phase. This method increases the number of previously under-studied data points, ultimately leading to improved performance and accuracy.

The proposed method in this paper demonstrates a remarkable advancement in video information retrieval, offering the potential to significantly enhance video search and retrieval systems across a wide range of applications. Our findings make video search more accessible and valuable to a broad range of users.

2. Related Works

The field of video information retrieval has been the subject of extensive research in recent years, due to the growing demand for effective and efficient methods for searching and retrieving video content. A wide range of techniques and approaches have been proposed to address the challenges associated with video retrieval, including traditional methods based on feature extraction and machine learning, as well as more recent deep learning-based methods. The literature in this area is vast, encompassing a wide range of topics, including the representation and encoding of video data, the development of effective similarity measures, and the handling of temporal and 3D information in video data. In this literature review, we provide an overview of the current state-of-the-art in video information retrieval, highlighting key contributions and developments in the field, and pointing to similarities and differences with our work.

The foundation of seq2seq models is simple neural network encoder and decoder recurrent models, such as LSTMs (Srivastava et al., 2015) and GRUs (Cho et al., 2014). Studies on the hierarchy for these encoders show that the better encoder networks in the model should lead to better results (Zhang et al., 2016). The proposed model builds upon this foundation, further advancing the encoder networks to enhance the overall performance. Consequently, in this work, we improve the state-of-the-art in this area by developing a deep convolutional and recurrent model using recent developments in the related areas.

Transformer architecture has been increasingly popular in recent years. It was applied to understand video data and perform video retrieval. The transformer, initially proposed for language understanding tasks, is a powerful architecture capable of capturing long-range dependencies and handling sequential data. One of the key features of the transformer is the self-attention mechanism, which allows the model to weigh the importance of different parts of the input data when making predictions. Several studies have applied transformer-based architectures to video understanding tasks such as action recognition, video captioning, and video retrieval. For example, (Girdhar et al., 2019) proposes a transformer-based architecture for video action recognition. Similarly, (Im & Choi, 2022) propose a transformer-based model for video captioning. The transformer has proven to be particularly useful in video understanding tasks, where the model needs to understand the relationships between different frames in a video. Our proposed method builds on the strengths of the transformer and further improves upon it by utilizing bi-directional dynamic time-warping to enhance the accuracy of video retrieval.

Video Retrieval. There is a variety of retrieval tasks and definitions in the multimedia community concerning the video retrieval problem. These vary with respect to the degree of similarity that determines whether a pair of videos are considered related and range from Near-Duplicate Video Retrieval (NDVR) with a very narrow scope where only almost identical videos are considered positive pairs (Wu et al., 2007), to very broad where videos from the same event (Revaud et al., 2013) in Event Video Retrieval (EVR) or with the same semantics (Basharat et al., 2008) are labeled as related. In the copy detection problem, given a query video, only videos containing nearly identical copies of it should be retrieved. Similar videos from the same incident should be considered irrelevant in such a scenario. On the other hand, problems such as news-oriented retrieval have radically different needs. Deep learning methods have been applied to certain applications such as face video retrieval (Choi & Kil, 2021) or as a general method (Kordopatis-Zilos et al., 2017). As an extension to the 3D case, (Deng et al., 2022) have proposed a pipeline to handle the 3D video data for retrieval purposes. However, there does not seem to be a strong consensus among researchers about the cases where the videos are all unlabeled, and the task is to retrieve videos sharing scenarios and semantic meanings. While there are a variety of retrieval tasks and definitions in the multimedia community, the proposed approach is a unified framework that can handle both 2D and 3D cases, therefore we focus on maximizing accuracy in video retrieval tasks where the temporal structure of the data is complex and multidimensional.

Convolutional Neural Networks (CNN's). CNNs (Karpathy et al., 2014) have been successfully applied to many tasks related to similarity video search (Zou et al., 2012; Bazzani et al., 2011; Mobahi et al., 2009). Unlike Deep Neural Networks, CNNs effectively exploit the structural locality in the spectral feature space. CNNs use filters with shared weights and pooling to give the model better spectral and temporal invariance properties. It typically generates more stable features compared to DNNs. The recent deep CNNs also show their superiority in image-related tasks compared to previous CNNs. Deep CNNs have been used for many tasks relating to video, such as predicting the next frame (Palm, 2012), learning invariant features from video (Chen et al., 2010; Srivastava et al., 2015) or video classification (Karpathy et al., 2014). Many training and modeling tricks, such as Residual Network (Kim et al., 2017), have been developed to enable training for such deep networks. While CNN's (Karpathy et al., 2014) have been applied to many tasks related to similarity video search, the proposed model takes advantage of bi-directional dynamic time-warping and recurrent neural network to better capture the temproal nature of the videos and improve the retrieval process.

Traditional Image Retrieval has been extensively explored as query by example or near-duplicate detection with high potential for the medical community (Rui et al., 1999; Ritendra et al., 2008). In literature, many research publications in image retrieval are based on non-deep learning methods. A competition for image-based retrieval was organized between 2004 and 2013. This case differs from the one addressed in this work because they were defined with 1-7 sample images accompanied by text. In the 2013 edition (Garcia Seco De Herrera et al., 2014), the best textual run by (Herrera et al., 2015) achieved the same performance as the best technique using both textual and visual features. As in previous years, visual-only approaches achieved much lower performance than textual and multimodal techniques. The best visual-based solution (Ozturkmenoglu et al., 2013) is based on the Color and Edge Directivity Descriptor (CEDD), a fuzzy color and texture histogram, and a Color Layout Descriptor. Content-based image retrieval in the medical domain has been addressed from low-level wavelet-based visual signatures (Quellec et al., 2010) to high-level concept detectors (Rahman et al., 2011). (Kalpathy-Cramer & Hersh, 2010) exploit using visual features to generate automatic text descriptors with computer vision algorithms and use these labels to support text-based queries. In contrast to most works above, we address the problem of video retrieval instead of still images. This venue has been previously explored in the literature. Moreover, (Quellec et al., 2011) presents a framework to retrieve short videos in real-time by modeling the motion content with a polynomial model. However, these methods are designed explicitly with specific domain knowledge. The proposed model extends these techniques to the video retrieval problem and further enhances them with bi-directional dynamic time-warping.

Auto-encoder has a long history of pre-training artificial neural networks (Ballard, 1987) and is widely used in recent models (Yoshua et al., 2009). Although this concept is rarely used by other deep learning models in this area, we found it to be fundamentally important for our semi-supervised learning purpose. In the proposed approach, we leverage this concept for semi-supervised learning in video retrieval tasks.

Representation Learning. Much of computer vision is about learning the representation, such as learning high-level image classification (Russakovsky et al., 2015), object relationships (Meister et al., 2018), or point-wise correspondences (Bansal et al., 2018; Kanazawa et al., 2016; Ce et al., 2010). However, there has been relatively little work on learning representation for aligning the content of different videos. In this work, we identify similar videos in the dataset, which is essentially aligning the content of two videos in a self-supervised manner, and do the automatic alignment of the visual data without any additional supervision. Although much of computer vision involves learning representation, we extend this notion to align the content of different videos in a self-supervised manner, thanks to the proposed bi-directional dynamic time-warping method.

ConvLSTM. (Shi et al., 2015) introduce convolutional LSTM (ConvLSTM) as an extension to the original LSTM, where the inner product is replaced by convolution operation in both input-to-state and state-to-state transitions. ConvLSTM effectively maintains the structural representations in the output and cell state. It has been shown to be a better tool than a fully connected LSTM layer for maintaining structural locality and more prone to overfitting. Besides, it reduces the number of parameters within the layer and enables potential more computations for better generalization. The ConvLSTM introduced by (Shi et al., 2015) has been a substantial tool in maintaining structural locality and preventing overfitting. However, in the research, we utilize the bi-directional dynamic time-warping approach, offering a more comprehensive temporal match and enhancing the video retrieval accuracy.

3. Model

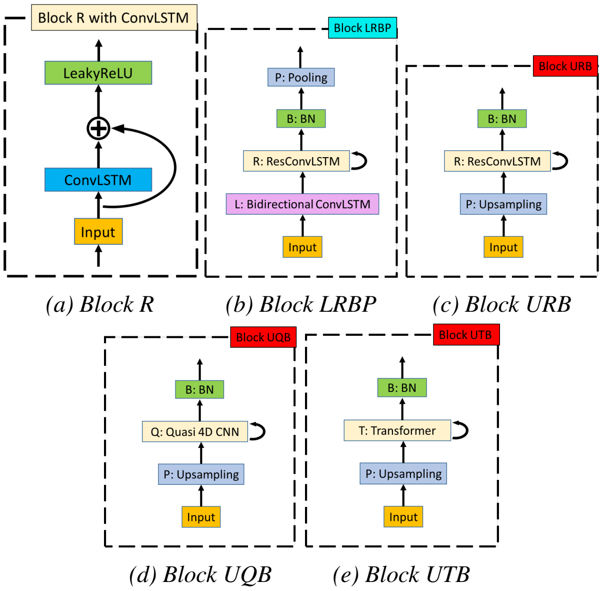
This paper unveils a different deep learning model tailored for both 2D and 3D similar video retrieval, marking advancements in the field. Our model is founded on a customized autoencoder structure, uniquely incorporating Convolutional LSTM (ConvLSTM), residual connections, and transformer blocks to form an efficient 2D sequence-to-sequence autoencoder.

Within the presented architecture for 2D video retrieval, it is imperative to discern between the established methodologies and novel introductions that are part of our research contribution. Starting off, Block R, which utilizes the ConvLSTM layer, is a traditional approach and is not new. In contrast, we are the first one to propose the LRBP block. The rationale behind adopting a bi-directional version of the ConvLSTM in the LRBP block is to adeptly capture the temporal dynamics in video data, especially those that might possess a reversible nature. Our empirical findings corroborate that the LRBP block manifests a superior performance relative to its non bi-directional counterpart, which is epitomized by the URB block. While the URB and UQB blocks remain a typical technique in employing ConvLSTM for video data, we introduce the amalgamation of the Quasi 4D CNN within an autoencoder structure through the UQB block. This innovation holds promising potential in transforming how visual temporal structures are perceived. Meanwhile, the UTB block, which leverages the transformer for embedding processing, does not offer a fresh perspective within our framework. We are the first to combine the blocks LRBP and UQB in this form.

3.1. 2D Seq2seq Autoencoder

This paper proposes a 2D sequence-to-sequence autoencoder for similar video retrieval. We first define several basic neural network modules as blocks to simplify the explanations. Block R, shown in Figure 2a, consists of a convolutional long short-term memory (ConvLSTM) layer with a residual connection and a LeakyReLU activation function. The LRBP block, defined in 2b, connects a block R with a pooling layer and a batch normalization layer. The encoder uses this block to compress the input into a lower dimension. The URB block, defined in 2c, connects an upsampling layer with block R and a batch normalization layer. The decoder uses this block to restore the embedding to a higher dimension. There is no residual connection between the layers of the encoder and decoder, which ensures the embedding vector after the encoder is a representation of the input video. To improve the generalization of the proposed model and enhance training speed, we include a residual connection within each block of LRBP and URB.

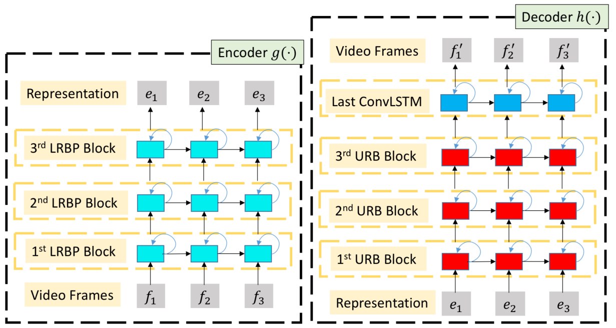
In addition to the URB block, we propose two other types of blocks to support the decoder: the UQB block and the UTB block. These blocks are used to handle the extra dimensionality in the 3D case. The UQB block replaces the ConvLSTM layer in the URB block with a Quasi 4D CNN layer, and the UTB block replaces the ConvLSTM layer with a transformer layer.



(a) Block R *(b) Block LRBP* *(c) Block URB*

(d) Block UQB *(e) Block UTB*

Figure 2. These figures show the model components for the 2D cases. (a) Shows the structure of block R. It contains a residual connected ConvLSTM layer and LeakyReLU activation function. (b) Shows the structure of block LRBP. (c) Shows the structure of block URB. (d) Shows the structure of block UQB. (e) Shows the structure of block UTB.



(a) The encoder *(b) The decoder*

Figure 3. (a) Shows the encoder architecture. (b) Shows the decoder architecture. The input video has three frames and the representation has three vectors.

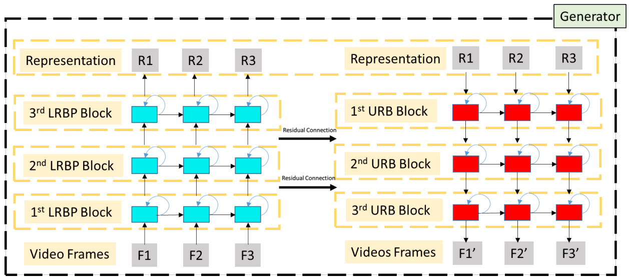


Figure 4. Illustrates how the encoder and decoder work together as an autoencoder, using LRBP and URB blocks as examples. The number of LRBP and URB blocks can be adjusted for better performance.

We trained the autoencoder to recover the original video and used the following loss function during training. In the autoencoder architecture, we have used a combination of the LRBP block in the encoder and the URB block in the decoder, as shown in Figure 3. Moreover, Figure 4 illustrates the comprehensive function of the proposed autoencoder architecture

$$L_{\text{autoencoder}} = \mathscr{L}\left(v, h_\tau(f_\theta(v))\right) \tag{1}$$

where $\mathbf{v}$ is an input video, $f_\theta(\cdot)$ is the encoder, $h_\tau(\cdot)$ is the decoder and $\mathscr{L}(\cdot,\cdot)$ represents a loss function. Hence $h_\tau(f_\theta(v))$ is the recovered video generated by the autoencoder.

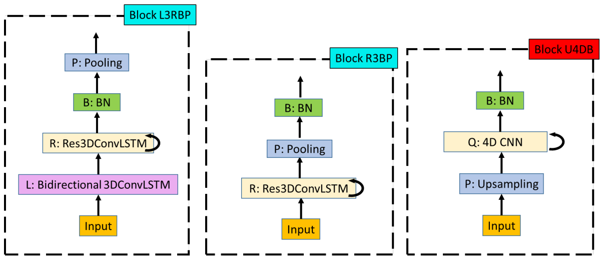
In our 2D video retrieval architecture, the LRBP and UQB blocks stand out as our primary contributions. The LRBP block employs an advanced bi-directional ConvLSTM, adeptly capturing intricate video dynamics. In contrast, the UQB block, incorporating its Quasi 4D CNN, brings forward a unique representation technique. This UQB innovation is not limited to 2D but will be expanded upon in 3D contexts. Collectively, these innovations significantly enhance video embedding, optimizing the retrieval process.

3.2. 3D Seq2seq Autoencoder

We also propose a 3D sequence-to-sequence autoencoder, an extension of the 2D sequence-to-sequence autoencoder model, which can be trained unsupervised for pretraining and fine-tuning with a supervised dataset. We are going to define three different variants for the proposed 3D model, namely M1-3D, M2-3D, and M3-3D.

In Table 1, we describe the 2D and 3D architectural differences among the six proposed 2D and 3D models. This block-based framework allows flexibility in the architecture and enables us to experiment with different combinations of blocks to achieve the best performance for similar video retrieval. The major difference between these models is: M1 is the 2D baseline model, M2 adds a Quasi 4D CNN layer in the decoder, M3 adds a transformer unit in the decoder; M1-3D is the 3D baseline model using a 3D ConvLSTM cell in the autoencoder's encoder and decoder parts. This allows the model to effectively extract the important features from the 3D video data and convert them into a compact and informative embedding for similar video retrieval. The second variant, M2-3D, replaces the 3D ConvLSTM cell with a Quasi 4D CNN layer for handling the 3D video reconstruction. The last variant, M3-3D, uses a transformer unit in the decoder.

For the 3D models, a series of unique building blocks are employed: L3RBP, R3BP, and U4DB. The L3RBP is our signature contribution, introducing a bi-directional connection to the 3D ConvLSTM. This innovative structure is tailored



(a) Block L3RBP *(b) Block R3BP* *(c) Block U4DB*

Figure 5. These figures show the model components for the 3D cases. (a) Shows the structure of block L3RBP. (b) Shows the structure of block R3BP. (c) Shows the structure of block U4DB.

Table 1. The blocks used in the encoder and decoder of all tested models

MODEL NO.	2D/3D	ENCODER	DECODER	DIFF
M1	2D	3 LRBP	3 URP	2D BASELINE
M2	2D	3 LRBP	3 UQP	+ QUASI 4D CNN
M3	2D	3 LRBP	3 UTP	+ TRANSFORMER
M1-3D	3D	3 L3RBP	3 R3BP	3D BASELINE
M2-3D	3D	3 L3RBP	3 U4DB	+ QUASI 4D CNN
M3-3D	3D	3 L3RBP	3 UTP	+ TRANSFORMER

explicitly for video inquiry, and it stands as a natural 3D progression of the LRBP block that we pioneered for 2D scenarios within this paper. On the other hand, the R3BP block harnesses the 3D ConvLSTM layer in a manner that aligns more with conventional techniques of video data processing. The U4DB block, conceptualized as the 3D iteration of the UQB block, primarily hinges on the 4D CNN layer. While the underpinning 4D CNN mechanism has been explored by others in the realm of video data, our approach offers a unique blend that synergizes seamlessly with our overarching model architecture. We are the first to combine the block L3RBP in such a form.

In the realm of 3D video retrieval, the new models introduce innovative adaptations that advance the field in two significant ways. Firstly, compared to 2D models, the 3D models harness the added depth dimension of videos to provide richer and more accurate representations of their content. This is achieved via a novel incorporation of 3D Convolutional LSTM cells, 3D CNN and Quasi 4D CNN layers into the autoencoder's encoder and decoder structure. The adaptation captures temporal correlations across frames more efficiently, leading to a more informative embedding for video similarity retrieval.

The second innovation is a distinct approach in dealing with the challenges posed by other state-of-the-art 3D video retrieval methods. While many existing techniques struggle to maintain performance with increasing video dimensionality, the proposed models, especially M2-3D and M3-3D, demonstrate robustness in handling higher-dimensional data. M2-3D substitutes the 3D ConvLSTM cell with a Quasi 4D CNN layer, introducing a novel way to manage 3D video reconstruction. The M3-3D model introduces a transformer unit in the decoder, a significant leap that helps model long-range temporal dependencies, offering superior performance in similarity retrieval tasks.

Conclusively, these enhancements furnish our 3D models with the capability to set pioneering standards for similar video retrieval. Their robust architecture and performance not only outshine the 2D models but also firmly position them at the forefront, rivaling other contemporary methods in the domain.

4. Algorithm

This section presents the problem setting of our video embedding learning problem. We define a distance metric for video embeddings that a triplet loss can train. We then propose the bi-directional dynamic time warping algorithm to convert the embedding into a distance metric.

4.1. Problem Setting

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

$$D(f_\theta(q), f_\theta(p)) = \|f_\theta(q) - f_\theta(p)\|_2^2 \qquad (2)$$

where $f_\theta(\cdot)$ is the embedding function that maps a video to a point in a Euclidean space, and θ are the system parameters. Additionally, we define a pairwise indicator function $I(\cdot,\cdot)$, specifying whether a pair of videos is near-duplicated. Formally, $I(q,p) = 1$ if q, p are NDVs (near-duplicate videos) and 0 otherwise.

The proposed model's objective is to learn an embedding function $f_\theta()$ that assigns smaller distances to similar video pairs compared to non-similar ones. Given a video with feature vector v, a similar video with $v+$ and a dissimilar video with $v-$, the embedding function $f_\theta(\cdot)$ should map video representations to a common space R^d, where d is the dimension of the feature embedding, in which the distance between query v and positive $v+$ is always smaller than the distance between the query v and negative $v-$:

$$D(f_\theta(v), f_\theta(v^+)) < D(f_\theta(v), f_\theta(v^-)), \qquad (3)$$

where $I(v, v^+) = 1$ and $I(v, v^-) = 0$ for all v, v^+ and v^-.

4.2. Triplet Loss

We use triplet loss to train the model and learn the above distance mapping function as a neural network. We define a collection of N training instances in the form of triplets $T = \{(v_i, v_i^+, v_i^-), i = 1, ..., N\}$ where v_i, v_i^+, v_i^- are feature vectors of a video, a similar positive video clip, and a negative video clip. A triplet expresses a relative similarity order among the three videos. We define the following hinge loss function for a given triplet:

$$L_\theta(v_i, v_i^+, v_i^-) = \\ \max\{0, D(f_\theta(v_i), f_\theta(v_i^+)) - D(f_\theta(v_i), f_\theta(v_i^-)) + \tau\} \qquad (4)$$

where τ is a margin parameter to ensure a sufficiently large difference between the positive and negative distances. This margin parameter also affects how the model is penalized if there is a violation for the desired triplet distance property. Finally, we use batch gradient descent to optimize the objective function described as triplet loss

$$\min_\theta \sum_{i=1}^{N} L_\theta(v_i, v_i^+, v_i^-) + \lambda \|\theta\|_2^2 \qquad (5)$$

where λ is a regularization parameter to prevent over-fitting of the model, and N is the total size of a triplet mini-batch. This triplet loss is also visualized in Figure 6. Minimizing this loss should narrow the query-positive distance while widening the query-negative distance, and thus lead to a representation satisfying the desirable ranking order. With an appropriate triplet generation strategy in place, the model should eventually learn a video representation that improves the effectiveness of the relevant video retrieval solution.

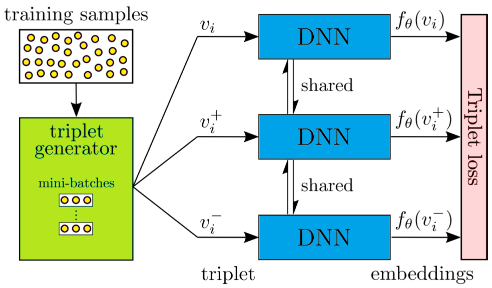


Figure 6. Illustration of a triplet loss in the proposed framework.

Algorithm 1 Training process for proposed model

```
   Input: Training data $\{v_i | i \in X\}$
1: Pre-train the autoencoder large video dataset from scratch with autoencoder loss (1) on $\{v_i | i \in X\}$.
2: Train the autoencoder with triplet-loss (5) on the training set $\{(v_i, v_i^+, v_i^-) | i \in \bar{X}_c\}$, where $v_i$ and $v_i^+$ are from the same classes and $v_i^-$ is from a different class. Here $\bar{X}_c$ is a multiset with respect to $X_c$.
3: Compute the similarity measure (2) for the training set. For each sample $i \in X_c$, find the top 20% training samples with highest $L_\theta(v_i, v_i^+, v_i^-)$ values. Let us denote these challenging samples as $\{(v_i, v_i^+, v_i^-) | i \in \tilde{X}_c\}$.
4: Fine tune the autoencoder with triplet-loss (5) and train with 50% of samples from the challenging sample sets $\tilde{X}_c$ and 50% of samples from the general training sets $\bar{X}_c / \tilde{X}_c$.
```

4.3. Training Scheme

The proposed retrieval method assumes that an embedding from different domains is distant from each other in the embedded space. Figure 7 shows how the trained encoder can map queries to embeddings from different content into separate neighborhoods in a T-SNE plot.

We assume we have a training set of videos $\{v_i | i \in X\}$ together with object classification already performed on a subset $X_c \subset X$. To this end, each $\{v_i | i \in X_c\}$ has the corresponding class.

As described in Algorithm 1, we use a four-step training scheme to optimize the performance of the proposed autoencoder-based video retrieval model. In step 2, a single sample $i \in X_c$ can yield multiple triplets and thus the need for $\bar{X}_c$. Step 4 intentionally increases the sample number of under-studied data to encourage the model to learn better on the difficult part of the dataset. The output of the algorithm is the semi-supervised trained encoder.

4.4. Video Similarity Search

Note that each video is represented as a sequence of frames or clips. We denote a video as v, and each frame as z_i, where $i \in M$ and M is the number of frames in the video. Alternatively, we can represent a video as a sequence of clips, denoted as $\mathbf{p}_i$, where $i \in N_x$ and $N_x = \lfloor \frac{M}{k} \rfloor$. The clips can be formed in two ways: disjoint consecutive frames or overlapping consecutive frames.

We then compute an embedded representation vector, v^p, for each clip of video v by mapping it to a vector using the encoder. We use a dynamic time-warping (DTW) method to measure similarity between two videos. This method is commonly used in time series analysis to measure the similarity between two temporal sequences. The DTW dis-

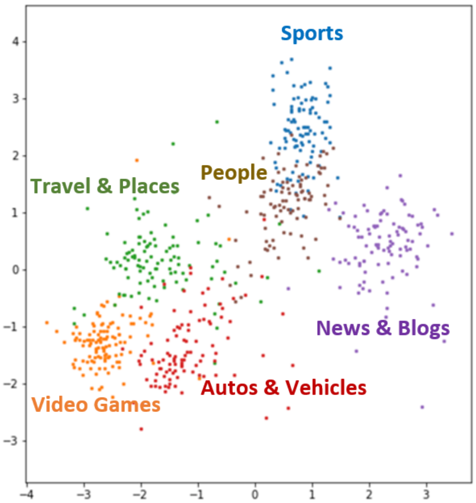


Figure 7. T-SNE plot that shows the trained encoder from model M3 maps embeddings of 6 random classes from CC_WEB_VIDEO dataset into separate neighborhoods.

tance between video clips v_1 and v_2 is solved as a dynamic programming problem, with the core state transfer formula being

$$D_{i,j} := \|v_1^i - v_2^j\|_2^2 + min(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}). \quad (6)$$

In order to provide a comprehensive solution to the video embedding matching problem, we expand on the traditional dynamic time warping (DTW) algorithm by proposing the bi-directional dynamic time warping (Bi-DTW) algorithm. Bi-DTW is distinguished by its ability to execute matching from both forward and backward directions, leading to an enhancement in the accuracy of the video retrieval system. The Bi-DTW algorithm computes both $DTW(v_1, v_2)$ and $DTW(\text{reverse}(v1), \text{reverse}(v2))$ where $\text{reverse}(v)$ is the sequence of clips in the reverse order. The output of Bi-DTW algorithm is $min(DTW(v1, v2), DTW(\text{reverse}(v1), \text{reverse}(v2)))$. Hence, Bi-DTW, with its dual-directional operation, not only enables us to determine the degree of similarity between two videos more effectively, but also enhances the ranking process of candidate videos. By comparing relevance from both the original and reverse sequences of the query video, it ensures a more comprehensive matching, thereby providing a superior and more versatile solution for the video retrieval task.

5. Numerical Experiments

In this section, we present experimental results, which demonstrate the effectiveness of the proposed method in terms of retrieval accuracy and computational efficiency when compared to the state-of-the-art methods.

5.1. Datasets

We have conducted numerical experiments on four different public datasets to evaluate the performance of the proposed model.

The four datasets used in the experiments are CCWebVideo, YouTube-8M-sub, s3DIS and Synthia-SF. The CCWebVideo dataset includes 12,790 videos and 27% near-duplicates. The YouTube-8M dataset is a large-scale video dataset which includes more than 7 million videos with 4,716 classes. The YouTube-8m-sub dataset we use is a random subset of YouTube-8m with around 100 videos from each class of YouTube-8m datasets. The Stanford 3D Indoor Scene (S3DIS) dataset contains 6 large-scale indoor areas with 271 rooms. Each point in the scene point cloud is annotated with one of the 13 semantic classes. The Synthia dataset is a synthetic dataset that consists of 9,400 multi-viewpoint photo-realistic frames rendered from a virtual city and comes with pixel-level semantic annotations for 13 classes. Synthia-SF is a subset of Synthia that only covers San Francisco. Table 2 summarizes the number of samples in each dataset. For the 3D datasets, we concatenate the depth information as an additional input dimension to the proposed 3D video auto-encoder.

Table 2. Descriptions for all four datasets.

Dataset	Type	No. of classes	No. of videos	Avg. video length
CCWebVideo	2D	24	533	265s
YouTube-8M-sub	2D	1000	100	3000s
S3DIS	3D	6	1600	251s
Synthia-SF	3D	6	417	83s

5.2. Implementation

We implemented the model using the Tensorflow 1.15 framework, and trained it on NVIDIA 3070 GPUs or equivalents. The model was trained with minibatch sizes of 32 and 8 clips for the 2D and 3D datasets respectively, using 4 GPUs in parallel. We employed the softmax loss function for the auto-encoder and applied L2 regularization of penalty ratio of 0.001 to the model's trainable parameters. These parameters were initialized using Xavier initialization.

The model was optimized using stochastic gradient descent (SGD) with an initial learning rate of 0.001 which decayed by 10 times every 10 epochs. Training ran for 50 epochs with early stopping after 5 epochs of no improvement. L2 regularization of 0.001 and momentum of 0.9 were used.

For the seq2seq auto-encoder, sequence-wise normalization was applied across multiple video sequences within each mini-batch. We computed the mean and variance statistics across all timesteps within this mini-batch for each output channel. Activation functions are ReLU.

For the encoder, it contains 3 LRBP layers and a dense layer. Each LRBP block took an input tensor of size 256x256x3. The BidirectionalConvLSTM layer employed a 3x3 kernel, a stride of 1, and had 64 hidden states. The subsequent ResConvLSTM used the same kernel size, padding, and stride but contained 32 hidden states. The output tensor after the residual connection was of size 256x256x96, which was then reduced to 128x128x16 after pooling. The output tensor was further reduced to 32x32x16 after the third LRBP block. This tensor was mapped to a 4000-entry embedding vector by a dense layer. In the Quasi 4D CNN, we used a 3x3x3 kernel. The transformer was applied to the embedding vectors, and consisted of 5 self-attention layers each with 3 attention heads. The transformer's hidden size was 512, and its intermediate size was 2048.

For the URB decoder, it contains three URB layers. The embedding vector was first transformed by a dense layer to 32x32x16 before entering the first URB block. Each URB block's ResConvLSTM utilized a 3x3 kernel, a stride of 1, and had 32 hidden states. The final ConvLSTM layer in the decoder had 3 filters, producing an output tensor of size 256x256x3.

The raw videos varied in length and resolution. We normalized them to 30 FPS, resized to 256 height keeping aspect ratio, cropped the center 256x256 region to get square RGB frames, and standardized the values to have 0 mean and unit variance across all videos. This standardized the data in terms of frame size, rate, and value ranges to effectively train deep learning models on the dataset.

5.3. Benchmarks

For the 2D video retrieval experiments, we compare the proposed models with a non-deep learning-based baseline method, which uses Scale Invariant Feature Transform (SIFT) to extract features from each frame in a clip and measures similarity between query clips and candidate clips using a bag of words method. Additionally, we also compare the proposed models to the state-of-the-art deep learning-based benchmark algorithm, NDVR-DML (Kordopatis-Zilos et al., 2017), which utilizes a Convolutional Neural Network and a Deep Metric Learning (DML) framework to generate discriminative global video representations and approximates an embedding function for accurate distance calculation between near-duplicate videos.

For the 3D video retrieval experiments, we use an autoencoder as the benchmark model, with 3D CNN layers as the encoder and decoder. This 3D autoencoder is a deep learning-based model that effectively extracts important features from the 3D video data and converts it into a compact and informative embedding for similar video retrieval. The architecture of this 3D autoencoder is similar to the proposed 3D models, with the main difference being the use of 3D CNN layers as opposed to the 3D ConvLSTM cell, Quasi 4D CNN layer, or transformer unit used in the proposed models.

5.4. Evaluation Metrics

For evaluation, we use Mean Average Precision (mAP) to measure the quality of the retrieval results. Measure mAP is a commonly used evaluation metric in information retrieval. It calculates the average precision of the retrieved results for a given query and is based on a binary relevance scale of 0 (non-relevant) or 1 (relevant).

The mean average precision we used is defined as in (Wu et al., 2007) and is formally stated as $mAP = \frac{1}{n}\sum_{i=0}^{n}\frac{i}{r_i}$, where n is the number of relevant videos to the query video, and r_i is the rank of the ith retrieved relevant video.

In order to evaluate the effectiveness of the proposed video retrieval methods, we carefully constructed a test set by randomly cropping the original training videos and concatenating the new clips. The test queries were formed by randomly selecting a set of frames from a video in the training set, with the original video serving as the query's ground truth. To ensure that the test queries differed from the database records, we designed the test set generation process to include consecutive frames and gaps between frames in the queries. This method of test set generation allows us to simulate real-world scenarios where the input query may be similar but not identical to the database records. The evaluation criteria include successful retrieval by clip and class, with the latter being a more stringent measure of performance.

5.5. Process

Training contains four steps, as described in Algorithm 1, which optimizes an autoencoder-based video retrieval model's performance. This schema includes pre-training the autoencoder on a large video dataset, training with a triplet-loss on a multiset derived from the training set, computing the similarity measure, and finally, fine-tuning with a second round of triplet-loss training on a mix of challenging and general training sets.

For inference, the dataset is split 70/15/15 into train, validation and test sets. Model efficacy is evaluated under two scenarios: by class and by clip. For class-level evaluation, validation clips are encoded into embeddings and compared against train embeddings. If retrieved training videos match the class of the query, precision is 1. For clip-level, precision is 1 only if the matched video contains the exact query clip. This matching process is repeated for all validation set videos and compared to the training set to compute metrics like mean average precision.

6. Results

In this section, we delve into the experimental results of our proposed models applied to 2D and 3D datasets, comparing their performances against various benchmarks. We use the metric of mean Average Precision to assess the models' effectiveness in video clip retrieval both by clip and class. Furthermore, a series of ablation studies help us understand the impact of different training techniques, loss functions, and the dynamic time warping method on the performance of our proposed models.

6.1. Results for 2D Datasets

We report the numerical results as mAP for the 2D datasets in Table 3. It shows that the proposed model has superior performance on the 2D datasets regarding the video clip retrieval accuracy by clip and by class.

Table 3. mAP results for 2D datasets

Model	CCWebVideo		Youtube-8m-sub	
	by class	by clip	by class	by clip
SIFT	58.8%	64.7%	27.9%	38.9%
NDVR-DML	84.4%	91.7%	**64.8%**	75.2%
M1	79.6%	85.3%	62.3%	71.8%
M2	81.7%	90.4%	65.4%	74.9%
M3	**85.2%**	**92.1%**	**64.8%**	**76.8%**

From Table 3, it can be seen that the proposed models M1, M2, and M3 all perform better than the non-deep learning-based SIFT baseline model, with M3 achieving the highest mean average precision across both datasets and both aggregation methods. Additionally, the models outperform the deep learning-based NDVR-DML benchmark model, with M3 achieving comparable or better results on all metrics. Overall, these results demonstrate the effectiveness of the proposed video retrieval models and their ability to achieve high levels of performance on various datasets and evaluation metrics.

For the CCWebVideo dataset using model M3, 14.8% of validation samples failed class-level retrieval. Upon inspection, around 32% of these failures retrieved videos of the wrong class but with visual similarity. For example, Fig 8 shows a comedy clip incorrectly matched to the music class. As evident in Figure 9, comedy and music clips can appear visually similar despite different semantic classes. This implies the encoder failed to sufficiently differentiate some inter-class nuances. Introducing the bidirectional dynamic time warping (Bi-DTW) method significantly boosted retrieval accuracy for certain classes. Animation clips improved from 92% to 98% class-level mAP, owing to Bi-DTW better handling symmetric motions of characters, as exemplified in Fig 10.

Overall, the analysis indicates room for improvement in handling inter-class visual similarities. Bi-DTW conferred notable gains for select classes by leveraging bidirectional temporal matching. Further inspection of embedding projections and retrieval results could provide additional insight into model limitations. Targeted sampling and training techniques may help differentiate challenging inter-class pairs.

Figure 8. Clip of Comedy class from CCWebVideo that is matched to Music class.

Figure 9. Typical clip from CCWebVideo Music class.

Figure 10. Example clip of Animation class from CCWebVideo.

6.2. Results for 3D Datasets

We report the numerical results as mAP for the 3D datasets in Table 4. It shows that the proposed model has superior performance on both 3D datasets about the video clip retrieval accuracy by clip and class.

According to Table 4, the performance of the proposed models varies across the two datasets and the different levels of aggregation. For example, on the S3DIS dataset, M1-3D and M3 perform similarly well in class-level mAP, with 61.4% and 61.1%, respectively, while M3 performs slightly better in terms of clip-level mAP with 78.1%. On the other hand, on the Synthia-SF dataset, M3 performs slightly better than the other proposed models in both class-level mAP and clip-level mAP.

The proposed models have not consistently outperformed the 3D autoencoder, except the model M3 is always better or on par with the baseline model. In addition, the table shows that the performance of the models is different when evaluated by class level or clip level. Figure 11 shows the training loss when we train the model with the triplet loss on the Synthia dataset. Based on these results, we

Table 4. mAP results for 3D datasets

MODEL	S3DIS		SYNTHIA-SF	
	BY CLASS	BY CLIP	BY CLASS	BY CLIP
3D AUTOENCODER	58.9%	77.9%	54.5%	82.5%
M1-3D	61.4%	75.4%	**55.6%**	82.1%
M2-3D	**62.3%**	73.3%	54.8%	81.9%
M3-3D	61.1%	**78.1%**	54.5%	**83.2%**

conclude that the models have great potential to improve the video retrieval results on 3D datasets and that it is important to consider the level of aggregation when evaluating the performance.

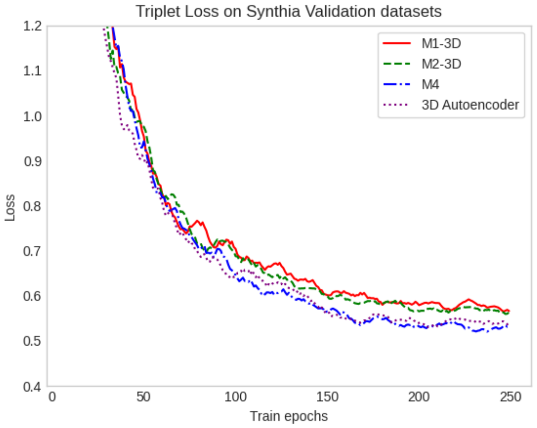


Figure 11. Training loss curve for proposed models

6.3. Ablation study: Effects of pretraining and challenging sample retraining

We investigate the impact of two key training techniques on the performance of the proposed deep-learning architecture. Specifically, we study the effects of pretraining on a large and diverse dataset and challenging sample retraining on the accuracy of similar video retrieval.

Firstly, we found that pretraining on a diverse dataset such as YouTube-8m-sub is crucial for improving the model's generalization ability and effectively learning from the given training set despite the complexity and intricacies of the video data. This required a significant investment of computation resources, with an approximate 24-hour pretraining period using 4 GPUs.

To validate the effectiveness of these techniques, we present an ablation study in Table 5 where we evaluate the performance of different variations of the baseline model, M3, on the CC_WEB_VIDEO dataset. The results show that the overall performance of M3 improves significantly when incorporating pretraining and training with challenging samples. Specifically, the model trained with pretraining and challenging samples in the second round achieved the best performance with 85.2% mAP by class and 92.1% mAP by clip. This suggests that pretraining and focusing on difficult samples during training can greatly enhance the model's performance for similar video retrieval.

Table 5. Effectiveness of pertaining and challenging sample retraining on CC_WEB_VIDEO dataset.

MODEL	CC_WEB_VIDEO	
	BY CLASS	BY CLIP
M3	54.1%	65.7%
M3 + 2ND PASS	65.5%	71.3%
M3 + CHALLENGING 2ND PASS	69.3%	78.5%
M3 + PRE-TRAIN	84.5%	89.2%
M3 + PRE-TRAIN AND CHALLENGING 2ND PASS	**85.2%**	**92.1%**

For the Synthia 3D dataset, Figure 12 shows an example query not improved by Bi-DTW, incorrectly matched to Figure 13 of a different class. Although many Synthia classes depend on filming time or weather, this failure stems from visual similarity of the street scenes. The Bi-DTW method may incorrectly relate clips appearing to drive in reverse on the same road despite different classes. Visually analogous roads filmed in opposing directions can be erroneously aligned by Bi-DTW. Further inspection into geometrically similar backgrounds causing inter-class confusion is needed. Additional constraints or tweaks to Bi-DTW's matching function may help mitigate false alignments from symmetric or reversed scenarios. Targeted data augmentation and sampling techniques could also help differentiate such challenging cases.

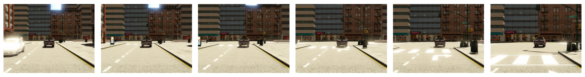


Figure 12. An example clip from Synthia.

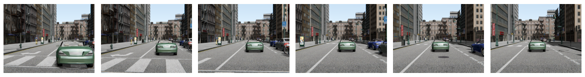


Figure 13. A clip of same location and different weather from Synthia.

6.4. Ablation study: Effectiveness of triplet loss

In this ablation study, shown as Figure 6, we evaluated the performance of different variations of the proposed model, M3, on the CC_WEB_VIDEO dataset. The results, as shown in the table, indicate that the incorporation of the triplet loss significantly improves the model's performance, achieving 54.1% mAP by class and 65.7% mAP by clip. Moreover, when pretraining is added to the model, the performance is further improved, achieving 84.5% mAP by class and 89.2% mAP by clip. These results suggest that the triplet loss is crucial for a similar video retrieval task, and pretraining enhances the model's generalization ability.

Table 6. effectiveness of triplet loss.

Model	CC_WEB_VIDEO	
	by class	by clip
M3 autoencoder	17.6%	32.2%
M3 autoencoder + triplet loss	54.1%	65.7%
M3 autoencoder + pre-train	29.6%	35.1%
M3 autoencoder + triplet loss + pre-train	**84.5%**	**89.2%**

6.5. Ablation study: Effectiveness of bi-directional DTW

In this ablation study, we evaluated the performance of the proposed model, M3, with two different dynamic time warping (DTW) methods: vanilla DTW and bi-directional DTW (Bi-DTW). The results presented in the table show that the overall performance of M3 improves significantly when utilizing the Bi-DTW method. Specifically, the model trained with Bi-DTW achieves 84.5% accuracy by class and 89.2% accuracy by clip level, whereas the model trained with vanilla DTW only achieves 72.1% accuracy by class and 85.4% accuracy by clip. This suggests that utilizing the Bi-DTW method dramatically enhances the model's performance for similar video retrieval.

Table 7. effectiveness of bi-directional DTW algorithm.

Model	CC_WEB_VIDEO	
	by class	by clip
M3 + vanilla DTW	72.1%	85.4%
M3 + Bi-DTW	**84.5%**	**89.2%**

Reflecting on the findings of our ablation studies, it becomes evident that the choice of methods hinges upon the specific scenario. For tasks necessitating precision and generalization, particularly in complex and intricate video data, it is recommended to incorporate pretraining and challenging sample retraining. These techniques significantly boost performance but require substantial computational resources, indicating their appropriateness in scenarios where such resources are available and accuracy is paramount. Meanwhile, the use of triplet loss has been proven to be effective across tasks, delivering substantial performance improvements. However, for tasks where even more enhanced results are sought, it is beneficial to couple the triplet loss technique with pretraining. When dealing with dynamic time warping, Bi-DTW stands out as a superior choice over the vanilla DTW, especially when striving for accuracy in similar video retrieval, making it the go-to method in such contexts.

7. Conclusion

In this paper, we proposed a novel approach for video information retrieval using a bi-directional dynamic time-warping method for measuring the similarity of video inquiries. We also introduced innovative ways to handle 3D video inquiry, including a 3D network architecture that extends the 2D video information retrieval deep network models to 3D video inquiry retrieval and a method for handling 3D video data as an extra depth layer. Additionally, we proposed a challenging sample retraining method to better address the challenges of training on video data. The experimental results on various public datasets, including CC_WEB_VIDEO, YouTube-8M-sub, S3DIS, and Synthia-SF, demonstrate the effectiveness of the proposed approach and its superiority over state-of-the-art methods.

The proposed models (M1, M2, M3) perform better than the non-deep learning (SIFT) and deep learning (NDVR-DML) baselines on the 2D datasets, with M3 achieving the highest mean average precision. On the 3D datasets, M3 always performed better or on par with the deep learning baseline model. The results also showed that the performance of the models varied when evaluated by class level or clip level and that it is important to consider the level of aggregation when evaluating the performance. Additionally, an ablation study was conducted to investigate the impact of pretraining on a large and diverse dataset and challenging sample retraining on the accuracy of similar video retrieval. The study found that both techniques were crucial for achieving high-performance levels on the video retrieval task.

It is important to acknowledge the variety and adaptability of the models that have been discussed in this paper. If the focus lies in handling 2D datasets, the proposed models M1, M2, and M3 clearly outperform both the non-deep learning and deep learning baselines, with M3 standing as the top achiever in terms of mean average precision on clips. For tasks involving 3D datasets, the M3 model consistently meets or surpasses the deep learning baseline model, suggesting its applicability for various dimensions of data. Hence, considering these diverse situations, the study suggests a careful selection of models according to the nature of the task. For instance, the M3 model is the preferred choice for both 2D and 3D datasets given its impressive results on clip mAP. If the dataset is 2D and the emphasis is per class mAP, then either M1 or M2 should be preferred.

It should also be stressed that the chosen model should ideally incorporate pretraining, challenging second round retraining, and bi-directional dynamic time warping, considering their significant contributions to enhancing performance in video information retrieval tasks.

The proposed approach not only provides a new method for video information retrieval but also opens up new possibilities for handling 3D video data more efficiently and accurately. The proposed difficulty rebalancing second round sampling technique is also a valuable addition to the field, as it addresses a common issue in training deep learning models on video data. The proposed approach can significantly impact the field of video information retrieval, and we look forward to further exploring its potential in future work.

References

Ballard, D. H. Modular Learning in Neural Networks. In *Proceedings of the sixth National Conference on Artificial Intelligence*, pp. 279–284, 1987.

Bansal, A., Ma, S., Ramanan, D., and Sheikh, Y. Recycle-GAN: Unsupervised Video Retargeting. In *Proceedings of the European Conference on Computer Vision*, pp. 119–135, 2018.

Basharat, A., Zhai, Y., and Shah, M. Content Based Video Matching Using Spatiotemporal Volumes. *Computer Vision and Image Understanding*, 110(3):360–377, 2008.

Bazzani, L., Freitas, N. D., Larochelle, H., Murino, V., and Ting, J.-A. Learning Attentional Policies for Tracking and Recognition in Video with Deep Networks. In *28th International Conference on Machine Learning*, pp. 937–944, 2011.

Ce, L., Yuen, J., and Torralba, A. SIFT Flow: Dense Correspondence Across Scenes and Its Applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010.

Chen, B., Marlin, B. M., and Ting, J.-a. *Deep Learning of Invariant Spatio-Temporal Features from Video*. PhD thesis, University of Columbia, 2010.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

Choi, Y. R. and Kil, R. M. Face Video Retrieval Based on the Deep CNN With RBF Loss. *IEEE Transactions on Image Processing*, 30:1015–1029, 2021.

Deng, R., Wu, Q., and Li, Y. 3D-CSL: Self-Supervised 3D Context Similarity Learning for Near-Duplicate Video Retrieval, November 2022.

Garcia Seco De Herrera, A., Kalpathy-Cramer, J., Demner-Fushman, D., Antani, S., and Müller, H. Overview of the ImageCLEF 2013 Medical Tasks. *Cross Language Evaluation Forum Workshop Proceedings*, 1179:219–232, 2014.

Girdhar, R., Joao Carreira, J., Doersch, C., and Zisserman, A. Video Action Transformer Network. In *2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, June 2019.

Herrera, A. G. S. d., Schaer, R., Markonis, D., and Müller, H. Comparing Fusion Techniques for the ImageCLEF 2013 Medical Case Retrieval Task. *Computerized Medical Imaging and Graphics*, 39:46–54, 2015.

Im, H. and Choi, Y.-S. UAT: Universal Attention Transformer for Video Captioning. *Multidisciplinary Digital Publishing Institute Sensors*, 22(13):4817, 2022. Publisher: MDPI.

Kalpathy-Cramer, J. and Hersh, W. Multimodal Medical Image Retrieval: Image Categorization to Improve Search Precision. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pp. 165–174, 2010.

Kanazawa, A., Jacobs, D. W., and Chandraker, M. Warpnet: Weakly Supervised Matching for Single-View Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3253–3261, 2016.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale Video Classification with Convolutional Neural Networks. *Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

Kim, J., El-Khamy, M., and Lee, J. Residual LSTM: Design of a Deep Recurrent Architecture for Distant Speech Recognition. *Interspeech*, pp. 1591–1595, 2017.

Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., and Kompatsiaris, Y. Near-duplicate Video Retrieval with Deep Metric Learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 347–356, 2017.

Meister, S., Hur, J., and Roth, S. UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Mobahi, H., Collobert, R., and Weston, J. Deep Learning from Temporal Coherence in Video. *26th Annual International Conference on Machine Learning*, pp. 737–744, 2009.

Ozturkmenoglu, O., Ceylan, N. M., and Alpkocak, A. The Effects of Modality Classification to Information Retrieval. In *Cross Language Evaluation Forum Workshop Proceedings*, pp. 1179, 2013.

Palm, R. B. *Prediction as a Candidate for Learning Deep Hierarchical Models of Data*. PhD thesis, Technical University of Denmark, 2012.

Quellec, G., Lamard, M., Cazuguel, G., Cochener, B., and Roux, C. Wavelet Optimization for Content-based Image Retrieval in Medical Databases. *Medical Image Analysis*, 14(2):227–241, 2010.

Quellec, G., Lamard, M., Cazuguel, G., Droueche, Z., Roux, C., and Cochener, B. Real-time Retrieval of Similar Videos with Application to Computer-aided Retinal Surgery. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4465–4468, 2011.

Rahman, M. M., Thoma, G. R., and K. Antani, S. A Learning-based Similarity Fusion and Filtering Approach for Biomedical Image Retrieval Using Svm Classification and Relevance Feedback. *IEEE Transactions on Information Technology in Biomedicine*, 15(4):640–646, 2011.

Revaud, J., Douze, M., Schmid, C., and Jegou, H. Event Retrieval in Large Video Collections with Circulant Temporal Encoding. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2459–2466, 2013.

Ritendra, D., Joshi, D., Li, J., and Wang, J. Z. Image Retrieval: Ideas, Influences, and Trends of The New Age. *ACM Computing Surveys*, 40(2):5, 2008.

Rui, Y., Huang, T. S., and Chang, S.-F. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, 1999.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

Srivastava, N., Mansimov, E., and Salakhutdinov, R. Unsupervised Learning of Video Representations using LSTMs. In *International Conference on Machine Learning*, pp. 843–852, 2015.

Wu, X., Hauptmann, A. G., and Ngo, C.-W. Practical Elimination of Near-Duplicates from Web Video Search. In *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 218, 2007.

Yoshua, B., Collobert, R., Weston, J., and Louradour, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Zhang, Y., Chan, W., and Jaitly, N. Very Deep Convolutional Networks For End-to-end Speech Recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

Zou, W. Y., Zhu, S., Ng, A. Y., and Yu, K. Deep Learning of Invariant Features via Simulated Fixations in Video. *Proceedings of Neural Information Processing Systems*, pp. 1–9, 2012.