Video action re-localization using spatio-temporal correlation

Akshaya Ramaswamy
TCS Research
Chennai, India
akshaya.ramaswamy@tcs.com

Karthik Seemakurthy
TCS Research
Bangalore, India
karthik.seemakurthy@tcs.com

Jayavardhana Gubbi
TCS Research
Bangalore, India
j.gubbi@tcs.com

Balamuralidhar P
TCS Research
Bangalore, India
balamurali.p@tcs.com

Abstract

Video re-localization plays an important role in locating the moments of interest in a long videos, and is critical for a variety of applications such as surveillance video monitoring and retrieving similar archived videos for further comparison and analysis. Current re-localization approaches compute a feature vector using a video query for each video frame, and explore various feature matching techniques. These features do not capture information from varying temporal windows, and the dimension reduction to a vector leads to loss of spatio-temporal context. For efficient feature comparison and matching among thousands of videos, we design a Siamese Spatio-Temporal network comprising Convolution Neural Network and Long Short-term Memory blocks (CNN-LSTM) for feature extraction, followed by a correlation layer for spatio-temporal feature matching. We extract video features at varying temporal scales, and localize one or more segments in the reference video that semantically match the query clip. Our approach is evaluated on two benchmark datasets: AVAv2.1- Search and ActivityNet-Search. We show an improvement of over 12% in the mean average precision compared to existing approaches. We perform ablation experiments and show that the modular architecture and the holistic feature extraction expands the scope of this work to multiple video search applications.

1. Introduction

With the rapid advancement in imaging and computing technologies, there is unprecedented growth in the amount of data that we generate every day. To cope with this explosion, proper organization is essential for on-demand retrieval of multi-media information. Visual data is unstructured and bulkier than text. As the database grows, the difficulty of efficient storage and retrieval of relevant search results increases. Traditional search engines index visual data based on the manual annotation of the surrounding metadata such as titles and meta-tags. There are research prototypes using deep learning techniques for automatically extracting details like actions, objects, and captions from videos. The similarity in this metadata can be further used to scrape relevant videos during a search. A downside to this approach is that textual descriptors are often inadequate to describe videos, simply because the same video can be described in different ways. Moreover, retrieval based on such meta-data as query yields too many results, making the search inefficient. With frequent adding and updating of multimedia in massive databases, it is also highly impractical to perform manual entry of all the attributes. Further, manual tagging of industrial videos is in its infancy. Content-based retrieval by using a video example as a query addresses many of these issues. It provides more flexibility and has the ability to query attributes such as texture or shape that are difficult to represent using keywords.

Video retrieval using a video query span multiple domains such as telemedicine, education, advertising and surveillance. Retrieval of semantically similar medical videos from archives can aid doctors in making an informed diagnosis. This can enable easier data sharing and enhance biomedical research. Retrieval of near-duplicate videos of trademarks is critical for copyright protection. In surveillance videos, locating key activity regions based on specific objects or events is crucial for quickly monitoring and narrowing down the search area.

Video retrieval using a video query has been attempted

before. The common approach is to compare the feature vector of a query video with that of each of the videos in the reference database [15, 27]. This approach computes similarity at the video level and does not localize the exact temporal region of the match in the video. Another major drawback is the reduction of entire video features to vectors, leading to a loss in information and also resulting in a large number of videos being retrieved. Video re-localization, which is also a type of retrieval, has recently seen a lot of traction. This task involves locating a smaller segment in a given reference video that matches the provided video query. This takes retrieval one step further and is very useful in quickly maneuvering through long videos and spotting the regions of interest.

Existing approaches for content-based video re-localization can be categorized into two main types: 1) similarity-based approach, wherein the features of the query and reference are compared at different temporal scales [29, 8]; 2) attention-based approach wherein the query features are fused with the frame-wise reference video features to get query-weighted attention features, and this is used to output the probability of a frame being a starting or ending frame [7, 11, 4]. A major drawback in both approaches is that the multi-scale spatio-temporal variations in the video are not exploited completely. While the first approach divides the videos into pre-determined segments, the second superimposes the entire query video into each frame to focus only on the relevant parts. A frame-wise feature similarity approach is suitable with natural language queries, but may not be ideal for video queries that are far more complex. This requires the design of a video matching network that can compare the objects in the reference video with those in the query video, and match features at multiple temporal scales.

Video re-localization can be used for other tasks like action counting and action sequence matching. These can be further extended to higher-level applications such as locating all interview clips from a news video or identifying an anomaly in an action sequence. The majority of the current approaches [7][13] work with video-level feature vectors and do not match features in the spatio-temporal space. This leads to a loss of spatio-temporal context and is highly unsuitable for fine-grained video search tasks. To address these deficiencies, we propose an approach that captures features at multiple time scales. This gives capability to match different types of actions and events. Such a framework will also be flexible and can be readily extended to multiple video search applications. The key contributions of the proposed approach in addition to generalization and robustness it achieves are as follows:

• In order to capture holistic information from the input videos, we extract features at multiple temporal scales such as spatial, micro spatio-temporal, and macro spatio-temporal levels.

• We introduce a novel patch-wise video correlation layer for spatio-temporal matching between the query and reference video features.

• We show that our network can be adapted to different temporal scales, thereby showing scope in multiple video search applications such as action counting, partial action matching and action sequence matching.

The rest of the article is structured as follows: section 2 presents the related work; section 3 covers the problem formulation, the architecture and explains in detail the main steps in the architecture; section 4 gives an overview of the datasets, followed by the experimental settings and results; section 5 presents the conclusion.

2. Related work

Content-based video search has been attempted in various ways using different modalities of queries. Examples include the use of natural language queries [26, 5, 18] and cross-modal embedding techniques [17, 22] for video retrieval, use of image as query [16, 1, 30] and use of video example as query [31, 14]. Video query-based video search can be posed as a retrieval task where all the videos relevant to the query are retrieved from a reference trimmed video database. It can be further specialized into a re-localization problem wherein one or more video segments matching the query are localized from a reference video. Approaches in literature either perform matching of global video features for the task of video retrieval or take up matching of spatial or spatio-temporal features for the task of re-localization. Our work deals with the problem of video re-localization with video examples as a query.

A number of datasets have been released in the video retrieval literature, and each one is designed for one or more of the following sub-classes: near-duplicate video retrieval [28], fine-grained video retrieval [12] and event video retrieval [21]. A general approach in the literature is to tackle only one of these retrieval problems [2, 28, 21]. This is highly inefficient and does not generalize well to retrieval based on multiple video attributes. Specific to *video re-localization*, the ActivityNet dataset [10] has been reorganized in [7]. Many of the prior works have used this for temporal action re-localization [11, 4, 7]. The common formulation is to predict the temporal boundaries of the reference segment(s) that match the action in the query video. This follows four steps: 1) feature extraction using pre-trained models such as C3D [24] or I3D [3] from each video, followed by feature aggregation; 2) attention based feature weighting in order to match the relevant part of the query with the reference; 3) feature matching; 4) mapping to the temporal boundaries of the segments that best match the query.

A lot of interest has gone into designing effective feature matching techniques. A cross-gated bilinear matching approach (CGBM) is developed in Feng [7] to encode the query-reference interactions at every timestep. A method called graph feature pyramid network (FPN) with dense predictions (GDP) is proposed by Chen *et al*. [4] where multi-scale co-attention is computed for the query-reference features, and this is followed by graph convolutions to encode the scene relationships. Weakly supervised approaches have also been proposed to make use of only video level information for query-reference pairs. Such methods make use of attention mechanism, and do not require the exact temporal location of the matching clip in the reference video. A weakly supervised multi-scale attention (MSA) approach is developed in [11], wherein, attention between multi-scale temporal features are used to estimate the similarity between the query video and the reference video. An attention-based approach is proposed by [23] to find the matching clip in reference using frame-level feature embeddings. The major drawbacks in the existing re-localization techniques are:
i) The extracted features are reduced to a vector using dimension reduction and feature aggregation. This results in the loss of key information that is critical in generalizing video search and retrieve operations. Although LSTMs are used in many cases [7] to incorporate long-range contextual information in the extracted features, that is not enough to counter the gap of spatial-temporal context as will be shown in the results. ii) Mapping the network to predict the start and endpoints of the matching reference segment is not robust to the general variability in actions being performed across different videos. This setting is also limiting in terms of the direct extension to derived applications like action counting, partial action matching, or action sequence matching. iii) Super-imposing the entire query video over the reference video frames using attention mechanism results in only parts of the query video getting focus. This step is deliberately performed to remove background clutter or irrelevant regions, but this results in sub-optimal matching especially for fine-grained action searches. Some of these issues are tackled in [13], where the ActivityNet-Search is evaluated for video retrieval. The output is mapped to the video-to-video-similarity, and this is computed using the frame-to-frame similarities.

More recently, the relatively difficult AVA dataset has been re-structured in [6] for spatio-temporal re-localization. Here, along with the query video, the corresponding action object location is also used to localize the matching video segment at the spatio-temporal level. The approach followed is similar to temporal re-localization with I3D-LSTM based feature extraction followed by attention-based query weighting. To localize at the spatio-temporal level, these blocks are incorporated in an R-CNN framework. Region proposals for the reference videos are generated and the feature matching and final similarity classification is done at the object level. A region proposal-based approach is also explored by [25] to remove interference from irrelevant regions in the video. Proposals are extracted from the reference video, and the query-reference feature matching is done using two blocks: attention-based fusion and semantic relevance measurement.

A key limitation in all these approaches targeting video re-localization is that the extracted features do not cover information captured across spatial and spatio-temporal scales. LSTMs have been used to capture long-term interactions, and multiple layer outputs have been tapped for multi-scale features, but matching at multiple temporal scales has not been explored. From the perspective of designing networks for video matching and retrieval, the difficulty is in effectively expressing a high-level semantic concept, such as a set of consecutive actions, with low-level visual features. Addressing the listed drawbacks and building on the advantages seen in previous works, we take the following steps: 1) extraction of multiple levels of spatio-temporal features from video, thereby enabling retrieval based on multiple attributes, 2) use of spatio-temporal correlation for effective matching of the video features, and 3) evaluation for the tasks of temporal re-localization and spatio-temporal re-localization

3. Problem formulation

We formulate the problem as re-localization of a set of actions performed by one or more objects of interest in the query video. Given a reference video $\mathbf{R} = \{\mathbf{r_1}, \mathbf{r_2}, ...\mathbf{r_m}\}$, query video $\mathbf{Q} = \{\mathbf{q_1}, \mathbf{q_2}....., \mathbf{q_n}\}$ and query object coordinates $\mathbf{BB^k} = \{\mathbf{bb_1^k}, \mathbf{bb_2^k}, ..., \mathbf{bb_n^k}\}$, our action re-localization system aims to identify one or more video segments in $\mathbf{R}$ that best describe the action performed by a query object of interest in the query video $\mathbf{Q}$. Here, $\mathbf{q_i}$ and $\mathbf{r_i}$ are i^{th} frames in the query and reference videos, respectively, $\mathbf{bb_i^k}$ indicates the bounding box coordinates corresponding to the k^{th} object of interest in the i^{th} query video frame. Here, we assume that the reference video is untrimmed, and the query video is trimmed ($m >> n$). This design can be extrapolated to other more complex matching requirements such as multi-object action re-localization or action sequence re-localization. The three inputs - query clip, reference clip and query objects - are given to our architecture which consists of four steps: 1) feature extraction, 2) region proposal generation and ROI pooling 3) featuring matching using spatio-temporal correlation and 4) bounding box refinement and similarity classification. The block diagram of the proposed architecture is shown in Figure 1.

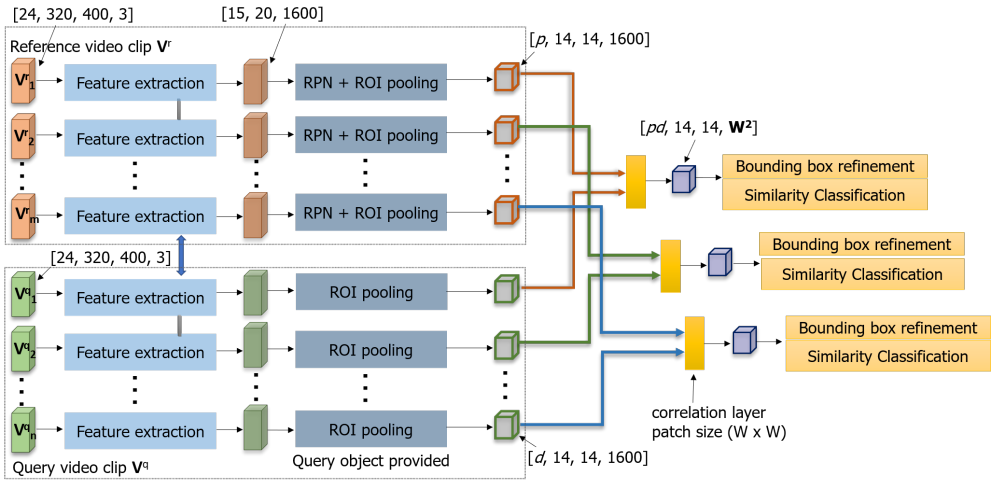

Figure 1. Network architecture for video re-localization using spatio-temporal correlation. 1) feature extraction, 2) region proposal generation and ROI pooling 3) featuring matching using spatio-temporal correlation and 4) bounding box refinement and similarity classification

3.1. Feature extraction

One of the key requirements for developing a generalized system to localize a wide variety of actions is to extract features at multiple levels efficiently. Motivated by the human visual system, a hierarchical architecture is introduced in our earlier work [19] for the task of spatio-temporal action detection and localization. Here, features are extracted in a pyramidal fashion, starting with a short timespan and building over it to cover longer time spans. Features are tapped at different time scales to capture and retain the layers of information contained in the video. We use this as our base feature extraction block and use a Siamese architecture to extract query features and reference features at three scales: spatial, micro spatio-temporal and macro-spatiotemporal.

Assuming the query video clip is trimmed and of length K seconds ($K >> 1$), and reference video is untrimmed ($M >> K$), the entire reference video is divided into overlapping K second clips. These clips are further divided into one-second subclips before feature extraction. Let $\mathbf{V_i^q} \in \mathbb{R}^{N\times H\times W\times 3}$ denote the i^{th} second subclip in the query clip and $\mathbf{V_{ki}^r} \in \mathbb{R}^{N\times H\times W\times 3}$ denote the i^{th} second subclip of the k^{th} clip in the reference video. Three sets of features are extracted from every query-reference subclip pair, and weights are shared across the Siamese sub-branch.

A clip spans K seconds, a sub-clip spans one second and a micro sub-clip spans one-third of a second; spatial features are extracted from a keyframe in every sub-clip, macro spatio-temporal features are extracted from each sub-clip and micro spatio-temporal features are extracted from each micro sub-clip. We show the process for a generic sub-clip $\mathbf{V_s}$. The same process is followed for both query sub-clips and reference sub-clips. Figure 2(C) shows the flow for the extraction of three sets of features from a sub-clip at 24 FPS.

To capture the spatial features, we select the center frame $\mathbf{I}$ in $\mathbf{V_s}$ as the keyframe and extract features using a ResNet-50 pretrained model [9]

$$\mathbf{S}_I = ResNet50\,(I) \tag{1}$$

where the dimension of $\mathbf{S}$ is $\mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times 1024}$.
$\mathbf{V_s}$ is split into 3 micro sub-clips and fed to the I3D pre-trained model [3] to extract micro spatio-temporal features. For a frame rate of twenty-four frames, $N = 24$ and each micro sub-clip will have eight frames. Let $\mathbf{V_s^p} \in \mathbb{R}^{\frac{N}{3}\times H\times W\times 3}(p = 1, 2, 3)$ be the three micro sub-clips given as input to a pre-trained I3D model. This is given by the following equation:

$$\mathbf{Tm_p} = \mathbf{I3D}(\mathbf{V_s^p}) \tag{2}$$

where $\mathbf{T_p} \in \mathbb{R}^{2\times\frac{H}{16}\times\frac{W}{16}\times 512}$ captures the micro spatio-temporal properties of the given input video micro sub-clip. We squeeze the output along the temporal dimension to reduce the features from four dimensions to three. This is done using a max operation along the first dimension to get the final micro spatio-temporal features for $\mathbf{V_s}$

$$\mathbf{Tm_p} = max(\mathbf{T_p}) \tag{3}$$

where $\mathbf{Tm} \in \mathbf{R}^{\frac{H}{16}\times\frac{W}{16}\times 512}$.
We concatenate the features of the three micro sub-clips, and squeeze it again along the temporal dimension as given below:

$$\mathbf{Tm} = concat(\mathbf{Tm_p}) \tag{4}$$

$$\mathbf{X_{V_s}} = max(\mathbf{Tm}) \tag{5}$$

where $\mathbf{X_{V_s}} \in \mathbf{R}^{\frac{H}{16}\times\frac{W}{16}\times 512}$ forms the aggregated micro sub-clip features of $\mathbf{V_s}$. We finally extract macro spatio-temporal features by making use of LSTM and attention

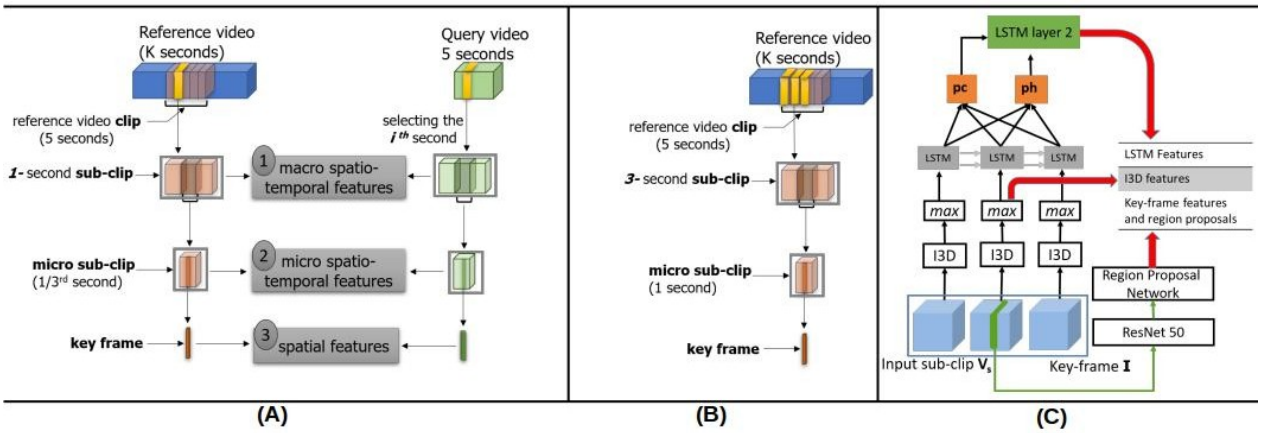

Figure 2. (A) Flow showing the different temporal scales for feature extraction at 24 FPS, (B) Adapting the feature extraction block for a 5 FPS video input, (C) Block diagram showing the extraction of three sets of features from an input video sub-clip

mechanism [19]. This follows a two layer LSTM architecture, one to capture and combine the information within each second and the other to capture long-term spatio-temporal variations across several seconds in time. The first LSTM layer takes each micro sub-clip feature $\mathbf{Tm_p}$ at every timestep. This is expressed by the following equation:

$$[\mathbf{c}_p, \mathbf{h}_p](\mathbf{V_s^p}) = LSTM_1\left(\mathbf{Tm_p}, [\mathbf{c}_{p-1}, \mathbf{h}_{p-1}]\right) \quad (6)$$

where $\mathbf{c}_p, \mathbf{h}_p \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times 64}, (p = 1, 2, 3)$ are the cell state and the hidden state of the LSTM module for the micro sub-clip $\mathbf{V_s^p}$. A key novelty in [19] is the use of an attention module to pool the latent hidden representations corresponding to each sub-clip. Let $\mathbf{ph}$ and $\mathbf{pc}$ denote the pooled hidden state and pooled cell state respectively, then

$$\mathbf{ph}(\mathbf{V_s}) = pool((\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3), \mathbf{X_{V_s}}) \quad (7)$$
$$\mathbf{pc}(\mathbf{V_s}) = pool((\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3), \mathbf{X_{V_s}}) \quad (8)$$

This first convLSTM layer captures the spatio-temporal information within the context of a sub-clip. The attention mechanism uses the aggregated micro spatio-temporal feature $\mathbf{X_{V_s}}$ given in equation (5) as context to pool the convLSTM states to get an output that spans one second. The pooled states are given to a second convLSTM layer to capture long-term spatio-temporal variations. The output of the second LSTM layer is given by the following expression:

$$\mathbf{hh}_t = LSTM_2\left([\mathbf{ph}_t, \mathbf{pc}_t], \mathbf{hh}_{t-1}\right) \quad (9)$$

where $\mathbf{hh}_t \in \mathbb{R}^{15\times 20\times 64}$ is the latent hidden representation of LSTM at second layer.

The features from $LSTM_2$, $ResNet50$ and $\mathbf{I3D}$ are concatenated to form the complete feature representation for each reference sub-clip and query sub-clip. Let $\mathbf{F_i^q} \in \mathbb{R}^{15\times 20\times 1600}$ and $\mathbf{F_{ki}^r} \in \mathbb{R}^{15\times 20\times 1600}$ be the outputs of feature extraction block then

$$\mathbf{F_{ki}^r} = [\mathbf{S}^r\ \mathbf{hh}_t^r\ \mathbf{Tm}^r]_{ki} \quad (10)$$
$$\mathbf{F_i^q} = [\mathbf{S}^q\ \mathbf{hh}_t^q\ \mathbf{Tm}^q]_i \quad (11)$$

The flow for the extraction of these three sets of features is shown in Figure 2(C). The query features $\mathbf{F_i^q}$ of each query sub-clip and the reference features $\mathbf{F_{ki}^r}$ of each reference sub-clip span multiple temporal scales, and are used further for proposal generation and matching clip localization.

3.2. Region proposal extraction

The region proposal network (RPN) [20] is applied only to the reference sub-clip features to generate proposals at every second. For the query clip, the object locations are already provided as input. Furthermore, the proposals are generated only for the keyframe at every second using the *ResNet50* features as shown in equation (1). By using the spatial features $[\mathbf{S}_r]_{ki}$, RPN generates P proposals. Let $\mathbf{P}_{ki}^r \in \mathbb{R}^{P\times 4}$ be the region proposals corresponding to the objects present in the reference video sub-clip $\mathbf{R}_{ki}$, then

$$\mathbf{P}_{ki}^r = RPN\left([\mathbf{S}_r]_{ki}\right) \quad (12)$$

The proposals in $\mathbf{P}_{ki}^r$ will be of different dimensions. Therefore, the features corresponding to the proposals regions will also vary in size. In order to compare the features of a proposal region in the reference sub-clip with the query object features, we need them to be of the same size. Following [20], we use Region of Interest (ROI) pooling operation to bring all the proposal features to same size. Here, we use the entire feature set $\mathbf{F_{ki}^r}$ shown in Eq. (10). Let $[\mathbf{F_{roi}}]_{ki}^r \in \mathbb{R}^{P\times 14\times 14\times 1600}$ denote the ROI aligned features after ROI pooling operation then,

$$[\mathbf{F_{roi}}]_{ki}^r = ROIpool\left(\mathbf{P}_{ki}^r, \mathbf{F}_{ki}^r\right) \quad (13)$$

Similarly, by using the query object bounding box $\mathbf{bb}_d^q$ of object $\mathbf{O}_d$ and query video clip features $\mathbf{F}_i^q$ we extract the object features as follows

$$[\mathbf{F_{obj}}]_i^q = ROIpool\left(\mathbf{bb}_i^q, \mathbf{F}_i^q\right) \tag{14}$$

where $[\mathbf{F_{obj}}]_i^q \in \mathbb{R}^{D\times 14\times 14\times 1600}$ The dimensions of $[\mathbf{F_{obj}}]_i^q$ depends on the number of objects D whose actions need to be re-localized in the reference video.

3.3. Feature matching using spatio-temporal correlation

Considering P proposals in the reference sub-clip and D objects of interest in the query sub-clip, we calculate the feature correlation between every proposal-query object pair. We compute a patch-wise correlation, which is given by the following equation for every spatial pixel location l_1 in the query feature $[\mathbf{F}_{\mathbf{obj}}^d]_i^q$ and l_2 in the reference feature $[\mathbf{F}_{\mathbf{roi}}^p]_{ki}^r$:

$$C_{dp}^{rq}(l_1, l_2) = \sum_c \sum_{l\in[-s,s]\times[-s,s]} \langle [\mathbf{F}_{\mathbf{obj}}^d]_i^q(l_1+l), [\mathbf{F}_{\mathbf{roi}}^p]_{ki}^r(l_2+l)\rangle \tag{15}$$

where $C_{pd}^{rq} \in \mathbb{R}^{14\times 14\times 14^2}$ is the correlation function, $(2s+1)$ x $(2s+1)$ is the spatial window at pixels l_1 and l_2 around which correlation is computed, and c spans the channel dimension. The correlation at a location l_1 can be computed with all locations l_2, but this requires a lot of computations. Hence, we restrict our region to a $W \times W$ neighbourhood around every pixel location. This gives us $C_{pd}^{rq} \in \mathbb{R}^{14\times 14\times W^2}$. Computing the correlation for every reference-query feature pair and stacking it along the first dimension results in correlation maps $C^{rq} \in \mathbb{R}^{PD\times 14\times 14\times W^2}$. The correlation maps capture the similarity between the actions in the query sub-clip and reference sub-clip. They are used in further refinement of the reference object bounding boxes and in the final classification to identify if the reference sub-clip matches the action in the query clip.

3.4. Bounding box refinement and classification

The correlation maps C^{rq} computed in the previous step are passed through block 3 of the ResNet model to extract a latent representation $\hat{C}^{rq} \in \mathbb{R}^{PD\times 14\times 14\times 512}$. This is further subjected to global average pooling as follows:

$$\mathbf{C}_{pool}^{rq} = GlobalPooling\left(\hat{C}^r\right) \tag{16}$$

where $\mathbf{C}_{pool}^{rq} \in \mathbb{R}^{PD\times 512}$ Following [20], we use two fully connected networks (FCN) to map each of the proposals to ground truth bounding boxes as well as perform binary classification into either *matching* or *not matching* with the query object. Let $\mathbf{W}_b$ and $\mathbf{W}_c$ be the weight matrices of FCN corresponding to bounding box and classification label respectively. The bounding box information and classification label can be computed by using the following expressions:

$$\mathbf{bb} = \mathbf{W}_b\mathbf{C}_{pool}^{rq} + \mathbf{b}_b \tag{17}$$

$$\mathbf{c} = \sigma\left(\mathbf{W}_c\mathbf{C}_{pool}^{rq} + \mathbf{b}_c\right) \tag{18}$$

where σ is the sigmoid operation, $\mathbf{bb} \in \mathbb{R}^{PD\times 4}$ and $\mathbf{c} \in \mathbb{R}^{PD\times 1}$ indicates the bounding boxes and the classification scores respectively. $\mathbf{b}_b$ and $\mathbf{b}_c$ are the bias vectors.

Let $\mathbf{c}^*$ be the ground truth labels such that c_i^* is 1 if the anchor is positive, else 0. As formulated in [20], $\mathbf{bb}^*$ contains the ground truth bounding box coordinates for the positive anchors.

We use the regression loss L_{reg} for bounding box refinement as defined in [20]:

$$L_{reg} = \sum_i c_i^* smooth_{L_1}\left(bb_i^*, bb_i\right) \tag{19}$$

The classification label is learnt by minimizing the following binary cross entropy loss:

$$L_{cls} = -\sum_i [c_i^* \log\left(c_i\right) + \left(1-c_i^*\right)\log\left(1-c_i\right)] \tag{20}$$

In this work, we train with one object of interest in the query video. So, $D = 1$ and the network outputs a set of bounding boxes with matching or no matching label. The sub-clips that have at least one matching object bounding box is assigned as a matching subclip for the provided query subclip. This can be extended to the multi-object matching scenario.

4. Experiments and results

We use two datasets - AVAv2.1 and ActivityNet - for evaluation of our approach.

AVAv2.1-Search: The original AVAv2.1 consists of 430 15-minute videos for the task of action detection and localization. It consists of ground truth bounding box annotations of persons and corresponding action labels in three categories: person pose, person-object interaction and person-person interaction. The fine-grained action annotation combined with dense action labeling makes this a very challenging dataset. To adapt it for video re-localization, this has been re-organized by [6] (AVAv2.1-Search) to form query-reference pairs. The train, validation and test splits have unique combined-action labels to be re-localized. The authors have used a spatio-temporal RCNN-based architecture with a warped LSTM and attention mechanism. To the best of our knowledge, this is the only prior approach that uses AVAv2.1 for video re-localization evaluation, and we compare our approach with this work (referred to as warpLSTM).

ActivityNet-Search: ActivityNet is an activity recognition dataset with more than 20,000 videos from 200 activity classes. This is re-arranged in [7] (ActivityNet-Search) such that the training set consists of videos from 160 classes, validation set and test set consists of 20 each. We use this split for the evaluation of the temporal re-localization task. Multiple approaches have used ActivityNet-Search for temporal re-localization. We compare our method with the following prior works - Context gating based bilinear matching (CGBM) [7], multi-scale attention (MSA) [11], attention feature matching (AFM) [23], graph feature pyramid network with dense predictions (GDP) [4] and semantic relevance learning network (SRL) [25].

4.1. Experimental Settings

The input to our video re-localization network is a query video and a reference video. In the training setup, we fix the query video length and reference video length to 5 seconds each. The actual query video length can be t_q seconds such that $2 \leq t_q \leq 5$. For $t_q \leq 5$, we pad the video with zero frames. For each training query video, we randomly crop a reference video of length 5 seconds. While testing, the query-reference pairs are fixed as per the testing split of the dataset and they can be of any length. For query and reference clips longer than the network input size, we take a window with an overlap of 4 seconds to cover the entire duration. The location of the first identified matching subclip is taken as the start time, and that of the last identified matching subclip as end time. To fix the patch-size W used in the correlation layer, we vary the size from 5 to 11 in steps of two. We choose an optimum value of 7×7, beyond which the performance does now show significant improvement. This is used in all our evaluation experiments.

Spatio-temporal video re-localization: For spatio-temporal video re-localization on AVAv2.1-Search, we extract frames at 24 FPS and compute the multi-level features as per 2(A). The final outputs are the actor bounding boxes and the binary classification at 1 FPS. ROI pooling is performed on the extracted features, and the ROI aligned features are correlated for final regression and classification. We use an Adam optimizer and a learning rate of 10^{-4}. We train the network on a 32GB Tesla GPU for 15000 epochs.

Temporal video re-localization: For temporal re-localization on ActivityNet-Search , the R-CNN framework is removed, and the features are directly correlated. The final output is a binary classification at 1 FPS. Previous approaches make use of I3D features extracted at 5 FPS. For fair comparison, we incorporate the same within our architecture with the following changes: i) we take 3-second sub-clips with 2-second overlap; and ii) at every second, the three sets of features are at frame-level, with 1-second span and with a 3-second span, respectively. This is visualized in figure 2(B).

4.2. Evaluation

For spatio-temporal video re-localization, we compute the mean average precision (mAP) metric for top-1 predictions at an IoU threshold of 0.5. Following Feng [7] for temporal re-localization, we compute the mAP of top-1 predictions at temporal IoU thresholds ranging from 0.5 to 0.9 at steps of 0.1.

4.2.1 Results on AVAv2.1-Search Dataset

The quantitative evaluation and comparison can be seen in Table 1.We achieve mAP of 42.15, a 13% increase over the current state-of-art approach warpLSTM. Two contributing factors to this are 1) the use of frame-level features for stable region proposals and 2) extraction of multi-scale spatio-temporal features to re-localize complex set of actions. Some example query video frames and corresponding re-localized frames are shown in Figure 3(B). The network output is visualized for two action classes: stand and answer the phone, and sit and hold an object. For accurate re-localization, an understanding of the action from varying temporal windows is required, and this can be observed in these actions, each having a different temporal dependence. We analyze the failure cases and observe poor performance in classes such as walking/running and playing music. One of the main challenges is large intra-class variations due to viewpoint, environment or visual appearance. Another difficulty especially in closely related actions like walk and run is in understanding the pace to differentiate between them.

Table 1. Quantitative evaluation on AVAv2.1-Search

Approach	mAP
warpLSTM [6]	29.1
Our approach	42.15

4.2.2 Results on ActivityNet

We evaluate our approach for temporal re-localization on the ActivityNet-Search dataset. We compute the mAP values at multiple IoU thresholds and compare them with the state-of-the-art, which is presented in Table 2. We also illustrate the qualitative results in Figure 3(A). Our approach results in a jump of 18% in terms of average mAP scores. This can be attributed to the fact that we use a large temporal neighborhood around each frame to compute relevant short-term and long-term features. As opposed to other methods, we do not perform feature pooling but compare the video features at the spatio-temporal level. This retains significant information and improves re-localization considerably.

Figure 3. Video re-localization outputs on (A) ActivityNet-Search for actions (top to bottom): hand car wash phone, and walk the dog, (B) AVAv2.1-Search for actions (top to bottom): stand and answer phone, and sit and hold object

Table 2. Quantitative evaluation on ActivityNet-Search

Method	tIoU					Avg. mAP
	0.5	0.6	0.7	0.8	0.9	
baseline [7]	24.3	17.4	12.0	5.9	2.2	12.4
AFM [23]	30.5	19.1	11.7	5.7	2.7	13.9
CGBM [7]	43.5	35.1	27.3	16.2	6.5	25.7
GDP [4]	44.0	35.4	27.7	20.0	12.1	27.8
MSA [11]	46.5	37.8	29.7	18.0	8.7	28.2
SRL [25]	40.6	40.5	40.4	30.0	16.1	33.5
Ours	**69.1**	**59.4**	**50.1**	**35.6**	**20.5**	**46.9**

Table 3. Ablation study in ActivityNet: 1) Feature concatenation (V_{concat}), and 2) Contribution of: i) spatial features ($V_{corr}|_s$), ii) spatial and micro spatio-temporal features($V_{corr}|_{ms}$); Comparison with best model: correlation with all features (V_{corr}). Highest mAP shown in bold, and least mAP in blue

Variations	mAP@1 — IoU 0.5	
	ActivityNet	AVA
V_{concat}	57.3	38.3
$V_{corr}|_s$	59.21	31.1
$V_{corr}|_{ms}$	65.8	36.82
V_{corr}	**69.1**	**42.15**

4.3. Ablation study

We perform ablation experiments on both the datasets to study the contribution of specific blocks in our architecture. 1) We evaluate the importance of the correlation layer by replacing it with concatenation. Instead of patch-wise correlation between the feature sets, we simply concatenate them, keeping all other settings same. As seen in Table 3, when compared with the best performing model (V_{corr}), the mAP drops by 11.8% on the ActivityNet-Search dataset. The correlation layer maximizes the feature matching, and the patch-wise computation ensures robust re-localization of the action in any part of the scene. The drop in mAP is not as significant in AVA, because the set of features are only from region proposals and not entire frames.

2) We evaluate two feature combinations: i) spatial ($V_{corr}|_s$); ii) spatial and micro spatio-temporal ($V_{corr}|_{ms}$). These are compared with the best model: correlation with all features (V_{corr}). Table 3 shows that the use of multi-scale features leads to an appreciable improvement in the re-localization performance. This is especially noteworthy in the AVA-Search performance. The AVA dataset consists of comparatively challenging actions that require temporal understanding for correct identification and matching.

Two key inferences can be made from the ablation study: the correlation layer plays an important role in temporal re-localization due to the patch-wise matching; secondly, the multi-scale feature extraction is essential for challenging actions that require temporal context and can be very useful for fine-grained action search.

5. Conclusion

We present a spatio-temporal correlation approach to re-localize an action in a reference video using a video query. We perform feature extraction from the video pair using a Siamese network, followed by feature matching and re-localization. Key contributions include multi-scale feature extraction with varying temporal windows and spatio-temporal feature matching using a correlation layer. We evaluate our network on two popular action re-localization databases: AVA-Search and ActivityNet-Search. We achieve an excellent improvement of over 12% in mAP on both the datasets. We demonstrate the adaptability of our network by employing different frame rates and video lengths. Our approach shows great promise in video re-localization, and the modular design enables straightforward extension to several video search applications.

References

[1] 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.

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

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[4] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *AAAI*, pages 10551–10558, 2020.

[5] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*, 2019.

[6] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Spatio-temporal video re-localization by warp LSTM. *CoRR*, abs/1905.03922, 2019.

[7] 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.

[8] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. MAC: mining activity concepts for language-based temporal localization. *CoRR*, abs/1811.08925, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[11] Yung-Han Huang, Kuang-Jui Hsu, Shyh-Kang Jeng, and Yen-Yu Lin. Weakly-supervised video re-localization with multiscale attention model. In *AAAI*, pages 11077–11084, 2020.

[12] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. FIVR: fine-grained incident video retrieval. *CoRR*, abs/1809.04094, 2018.

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

[14] 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.

[15] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep video hashing. *IEEE Transactions on Multimedia*, 19(6):1209–1219, 2016.

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

[17] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. New York, NY, USA, 2018. Association for Computing Machinery.

[18] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.

[19] A. Ramaswamy, K. Seemakurthy, J. Gubbi, and B. Purushothaman. Spatio-temporal action detection and localization using a hierarchical lstm. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3303–3312, 2020.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[21] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2459–2466, 2013.

[22] Didac Surís, Amanda Duarte, Amaia Salvador, Jordi Torres, and Xavier Giró-i Nieto. Cross-modal embeddings for video and audio retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 0–0, 2018.

[23] Haoyu Tang, Jihua Zhu, Zan Gao, Tao Zhuo, and Zhiyong Cheng. Attention feature matching for

weakly-supervised video relocalization. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, MMAsia '20, New York, NY, USA, 2021. Association for Computing Machinery.

[24] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[25] Ruolin Wang and Y. Zhou. A feature pair fusion and hierarchical learning framework for video relocalization. *IEEE International Conference on Image Processing (ICIP)*, pages 2341–2345, 2020.

[26] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 450–459, 2019.

[27] Gengshen Wu, Jungong Han, Yuchen Guo, Li Liu, Guiguang Ding, Qiang Ni, and Ling Shao. Unsupervised deep video hashing via balanced code for large-scale video retrieval. *IEEE Transactions on Image Processing*, 28(4), 2018.

[28] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. New York, NY, USA, 2007. Association for Computing Machinery.

[29] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. *CoRR*, abs/1804.05113, 2018.

[30] Ruicong Xu, Li Niu, Jianfu Zhang, and Liqing Zhang. A proposal-based approach for activity image-to-video retrieval. In *AAAI*, pages 12524–12531, 2020.

[31] Yuan Zhou, Mingfei Wang, Ruolin Wang, and Shuwei Huo. Graph neural network for video-query based video moment retrieval. *arXiv preprint arXiv:2007.09877*, 2020.