Spatio-temporal Video Re-localization by Warp LSTM

Yang Feng[♯*] Lin Ma[♮†] Wei Liu[♮] Jiebo Luo[♯]
[♮]Tencent AI Lab [♯]University of Rochester
{yfeng23,jluo}@cs.rochester.edu forest.linma@gmail.com wl2223@columbia.edu

Abstract

The need for efficiently finding the video content a user wants is increasing because of the erupting of user-generated videos on the Web. Existing keyword-based or content-based video retrieval methods usually determine what occurs in a video but not when and where. In this paper, we make an answer to the question of when and where by formulating a new task, namely spatio-temporal video re-localization. Specifically, given a query video and a reference video, spatio-temporal video re-localization aims to localize tubelets in the reference video such that the tubelets semantically correspond to the query. To accurately localize the desired tubelets in the reference video, we propose a novel warp LSTM network, which propagates the spatio-temporal information for a long period and thereby captures the corresponding long-term dependencies. Another issue for spatio-temporal video re-localization is the lack of properly labeled video datasets. Therefore, we reorganize the videos in the AVA dataset to form a new dataset for spatio-temporal video re-localization research. Extensive experimental results show that the proposed model achieves superior performances over the designed baselines on the spatio-temporal video re-localization task.

1. Introduction

Video sharing websites or APPs are becoming more popular than ever before. Besides the traditional video-sharing websites, including YouTube[1] and Facebook[2], the recently emerged short video sharing APPs, such as SnapChat[3] and TikTok[4], arouse the passion of ordinary users for creating and sharing video contents. With more and more videos generated every day, exploring so many videos becomes increasingly challenging. It is necessary to build tools which can help users find the video contents they want efficiently.

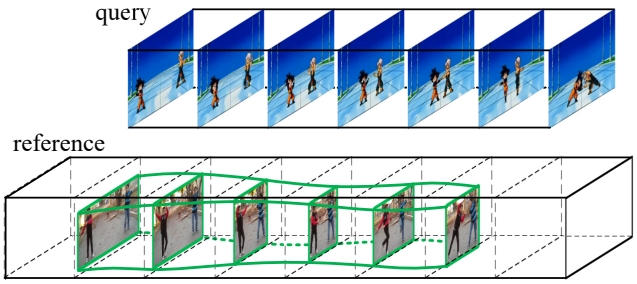

Figure 1. The query is a video containing an action performed by two characters. The reference video contains two boys performing the same action. Given the query, spatio-temporal video re-localization aims to localize the tubelets in the reference video such that the tubelets express the same visual concept as the query. The desired tubelet in the reference is marked by green. Best viewed in color.

Keyword-based video search is prevalent among users when they want to find some videos. Although it is a powerful method, keyword-based video search results are largely determined by the text information associated with the videos. As such, content-based video retrieval (CBVR) methods [2, 3, 5, 11, 23, 30, 44, 47, 51] are proposed for tackling this problem. With the indexing and retrieval techniques, a large list of videos is returned by the CBVR system. Only the top results will be viewed by a user, as it is time-consuming to browse the whole video from the beginning to the end and thereby determine the relevance.

Two kinds of methods are designed to avoid browsing the whole video. The first kind is video summarization methods [32, 58], which generate a short synopsis for a long video. The second kind of methods [7, 8, 13, 14, 19, 22, 31, 37, 41] try to trim the video segment of interest. Using natural language as a query, [14, 19] retrieve a specific temporal segment in a video, which shares the same semantic meaning as the query. By replacing the query sentence with a sample video clip, video re-localization [13] aims to temporally localize video segments, which semantically correspond to the query video clip.

In this paper, we extend the temporal video re-localization [13] to the spatio-temporal domain. Specif-

*This work was done while Yang Feng was a Research Intern with Tencent AI Lab.
†Corresponding author.
[1]https://www.youtube.com
[2]https://www.facebook.com
[3]https://www.snapchat.com
[4]https://www.tiktok.com

ically, given a query video, spatio-temporal video re-localization (STVR) aims to localize tubelets in a reference video such that the tubelets are semantically coherent with the query video. Figure 1 illustrates an example pair of query and reference videos. There are several advantages in localizing tubelets over temporal localization based on whole frames. First, localizing tubelets can handle the cases where multiple events are happening at the same time in the reference video. When using whole video frames for recognition or temporal detection tasks, it usually assumes that only one event is undergoing. The assumption rarely holds in unconstrained environments. Second, the recognition accuracy will substantially increase because the influence of background regions is reduced by only focusing on specific regions. STVR is also a more challenging task than temporal video re-localization. First, the training videos should be labeled with bounding boxes over a long period, which consumes more human labors than temporal annotation only. Second, detecting the bounding boxes at each frame is more difficult than only localizing the starting and ending boundary points.

To address the STVR task, we propose a matching framework consisting of three modules: query encoding, reference encoding, and query-reference interaction. The query encoding module encodes a query video of arbitrary resolution into a series of fixed size feature cubes. The reference encoding module encodes the given reference video in a different manner. To keep the detailed spatio information in the reference, the shape of the reference feature cube is proportional to the resolution of the reference video. In the query-reference interaction module, several bounding box proposals are generated for the reference and then each proposal is matched with the query to determine whether a proposal and the query are semantically corresponding to each other.

To accurately localize the tubelets in the reference, the long-term spatio-temporal information needs to be modeled. We propose a novel warp LSTM network for this purpose. Warp LSTM is a variant of ConvLSTM [39]. In ConvLSTM, the previous hidden state is concatenated with the current input for further computation. Different from ConvLSTM, the previous hidden state in warp LSTM is warped before the concatenation to make the previous hidden state be aligned with the current input if any movement in the video makes them unaligned. The warp of the hidden state at a previous time-step can compensate for small movements in the video, which accurately aggregates the spatio history information of moving objects.

In order to train the matching model for STVR, we create a new dataset by reorganizing the videos in the AVA dataset [16]. The AVA dataset is originally used for spatio-temporal action localization. Each action tubelet is annotated with one or several atomic action labels. We use one action tubelet as the query and find the tubelets with the same action labels in the reference video. Two action tubelets are semantically corresponding to each other if the action labels of the two tubelets are exactly the same. The AVA dataset provides a subset of videos for training and another subset of videos for validation. We further split the action tubelets into training, validation, and test subsets according to their action categories. Such a splitting guarantees that the validation and testing categories do not overlap with the training categories.

In summary, our contributions are four-fold:

- We make the first attempt to tackle the STVR task, which aims to localize tubelets in the reference video such that the tubelets semantically correspond to a given query video.
- We propose a novel warp LSTM network to propagate the spatio-temporal information between adjacent frames for a long period and thereby capture the corresponding long-term dependencies.
- We reorganize the videos in the AVA dataset [16] to form a new dataset for the research on STVR.
- We conduct extensive experiments on the new dataset, which shows that the warp LSTM performs better than the competing methods.

2. Related Work

Video Representations. Convolutional Neural Networks (CNNs) have broken many records of computer vision tasks, such as image classification [18, 36], object detection [24], semantic segmentation [9], facial expression recognition [55], and captioning [10, 28, 48, 49, 53]. Due to the great success of CNNs on images, many researchers tried to apply CNNs on videos. [34, 42, 54] are mainly based on 2D CNNs, in which the motion information is not fully exploited. 3D CNNs are proposed in [27, 46, 56] to capture more complex motion patterns. The recently proposed I3D feature [4] has achieved state-of-the-art action recognition results. Compared with 3D CNNs, the proposed warp LSTM is able to model the long-term spatio-temporal information of moving objects for classification and localization tasks by explicitly modeling the movements in videos.

Video Re-localization. Video Re-localization [13] aims to find segments in reference videos semantically corresponding to a given query video. A more specialized task, one-shot action localization [52], focuses on the temporal detection of actions in videos giving an example. The STVR task to be solved in this paper is an extension of temporal video re-localization. Besides predicting the starting and ending points of a video segment, STVR also detects the spatio localization of the video content that users are interested in. Hoogs *et al.* [21] designed a system to spatio-temporally retrieve people and vehicles in surveil-

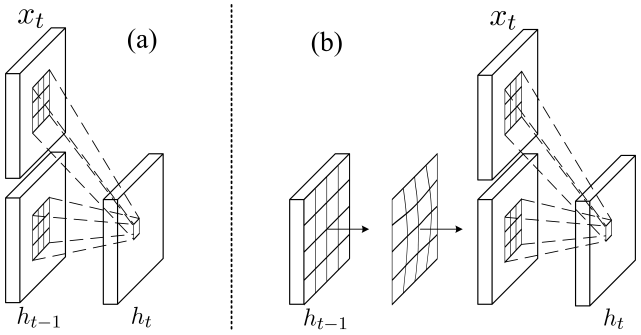

Figure 2. Comparison between ConvLSTM and warp LSTM. (a) In ConvLSTM, the input x_t and the hidden state at previous time-step h_{t-1} are convolved with different filters and then added to produce the new hidden state h_t. (b) In warp LSTM, h_{t-1} is warped by a differentiable spline interpolation before the convolution to compensate for the small motion between consecutive video clips.

lance videos. STVR is different in that it is not specialized in certain categories or types of videos.

Spatio-temporal Detection. Two related vision tasks are video object detection and spatio-temporal action detection. All the three tasks need to model long-term spatio-temporal dependencies to predict tubelets in videos. Although the temporal information inside a clip is considered in [22, 31], the bounding boxes are predicted independently of the frames outside the short clip. To solve this problem, both [33, 37] extract tubelet proposals from videos in the first stage and make the classification in the second stage. One assumption used in both [33, 37] is that the reception field of CNN features is large enough to handle the small movements in a short time. With this assumption, the feature cropped at a previous anchor location is used to predict the bounding boxes at the current frame. Although the reception field is large enough to cover the objects with small movements, the bounding box prediction will become a more difficult task on the feature map with offsets. Different from them, we align the previous feature maps with the current feature map by warping. The proposed warp LSTM can reduce the offset of the previous feature map and thereby reduce the burden of the bounding box prediction module.

3. Spatio-temporal Information Propagation

In this section, we present our proposed warp LSTM network for modeling the long-term spatio-temporal information in videos. Warp LSTM is a variant of ConvLSTM [39]. We first give the background knowledge of ConvLSTM.

3.1. ConvLSTM

ConvLSTM extends the fully-connected LSTM [20] to have convolutional structures in both the input-to-state and state-to-state transitions:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i),\\
g_t &= \sigma(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g),\\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f),\\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o),\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t,\\
h_t &= o_t \odot \phi(c_t),
\end{aligned}
\tag{1}
$$

where x_t, h_t, c_t, i_t, f_t, and o_t are the ConvLSTM input, hidden state, memory cell, input gate, forget gate, and output gate at time-step t, respectively. All the Ws and bs are the parameters of the ConvLSTM layer. $*$ is the convolution operation and $\odot$ is the element-wise product. σ and ϕ are sigmoid non-linearity and hyperbolic tangent nonlinearity, respectively. In Eq. (1), x_t and h_{t-1} are convolved with different filters and then added for later computation, as shown in Figure 2. This operation is equivalent to first concatenating x_t and h_{t-1} along the channel dimension and then computing the convolution.

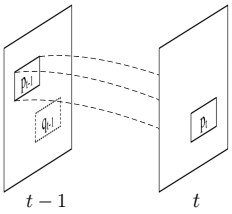

Figure 3. An illustration of a moving object in two consecutive time-steps. At time-step $t-1$, the object is located at the bounding box p_{t-1}. In the following time-step, the object moves to the location of bounding box p_t. q_{t-1} is a bounding box at time-step $t-1$ having the same position as p_t. Please note that the content in q_{t-1} may be not semantically related to the object in p_t.

3.2. Warp LSTM

If no movement happens in the video at the t-th time-step, the concatenation of x_t and h_{t-1} in ConvLSTM is perfectly fine. However, when motion happens at time-step t, the concatenation of x_t and h_{t-1} may cause errors in spatio-temporal localization tasks. Figure 3 shows a moving object at two consecutive time-steps. At time-step $t-1$, the object is located at the bounding box p_{t-1}. In the following time-step, the object moves to the location of bounding box p_t. q_{t-1} is a bounding box at time-step $t-1$ having the same position as p_t. The content in q_{t-1} may be depicting objects other than the aforementioned object. As such, simply concatenating the features at the locations of q_{t-1} and p_t may introduce noises into the classification and localization of p_t.

We propose warp LSTM to address this issue by warping the hidden state at the previous time-step before concatenating it with the input. Figure 2 illustrates the proposed warp

LSTM, where the warp is implemented by the differentiable spline interpolation [12], as illustrated in Figure 4. Given a set of 2-D control points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ on the hidden state h, the warp operation tries to shift (x_i, y_i) to a new position $(x_i + dx_i, y_i + dy_i)$, where n is the number of control points and (dx_i, dy_i) is the desired displacement of the i-th control point. Let h' denote the warped hidden state, then we have:

$$h'[x_i + dx_i, y_i + dy_i] = h[x_i, y_i], \forall i \in \{1, \ldots, n\}. \quad (2)$$

Besides shifting the control points, the warping is continuous on the whole 2D space of h, resulting in a dense flow field. The flow field is estimated by the polyharmonic interpolation [26]:

$$s(x, y) = \sum_{i=1}^{n} w_i \phi_k(\|(x, y) - (x_i, y_i)\|) + v_1 x + v_2 y + v_3, \quad (3)$$

where ϕ_k is a set of radial basis functions. w_i, v_1, v_2, and v_3 are interpolation parameters. After optimization, the polyharmonic interpolation s will shift the control points exactly to their desired locations. In addition, the warped h' is a differentiable function of h, (x_i, y_i), and (dx_i, dy_i).

In practice, the control points are fixed in advance. We evenly put horizontal lines and vertical lines in the 2D space of h and put control points on the intersections of horizontal and vertical lines. The displacement (dx_i, dy_i) is predicted by an additional convolutional layer. We also add extra control points with zero displacements at the boundary. Two radial basis functions, *i.e.*, $\phi_1(r) = r$ and $\phi_2(r) = r^2 \log(r)$, are chosen for the interpolation. The proposed warp LSTM is defined as:

$$\begin{aligned}
d_{t-1} &= W_{xd} * x_t + W_{hd} * h_{t-1} + b_d, \\
h'_{t-1} &= \text{warp}(h_{t-1}, d_{t-1}), \\
c'_{t-1} &= \text{warp}(c_{t-1}, d_{t-1}), \\
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h'_{t-1} + b_i), \\
g_t &= \sigma(W_{xg} * x_t + W_{hg} * h'_{t-1} + b_g), \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * h'_{t-1} + b_f), \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h'_{t-1} + b_o), \\
c_t &= f_t \odot c'_{t-1} + i_t \odot g_t, \\
h_t &= o_t \odot \phi(c_t),
\end{aligned} \quad (4)$$

where d_{t-1}, h'_{t-1}, and c'_{t-1} are the displacement, warped hidden state, and warped memory cell at time-step $t-1$, respectively. warp$(\cdot, \cdot)$ is the sparse image warping function[5], which warps an image based on control point displacements.

Discussion. The closest work to our proposed warp LSTM is TrajGRU [40], which also warps the feature map at a previous time-step to the current time-step. However, there are two major differences between the two methods. The motivation of TrajGRU is to learn a dynamic connection structure, *e.g.*, replacing the fixed 3×3 convolution with 5 learned dynamic links. Our motivation is to align the previous feature map with the current feature map. Several dense flow fields are predicted by convolutional layers in TrajGRU for warping, while the displacements of a set of control points are predicted in the warp LSTM. The warp computed by polyharmonic interpolation is continuous everywhere, while the dense flows generated by convolutional layers in TrajGRU may be not. TrajMF [29] is also designed for explicit motion modeling. Compared with TrajMF, the proposed warp LSTM is not handcrafted and is thus benefiting from the feature learning ability of deep neural networks.

[5] https://www.tensorflow.org/api_docs/python/tf/contrib/image/sparse_image_warp

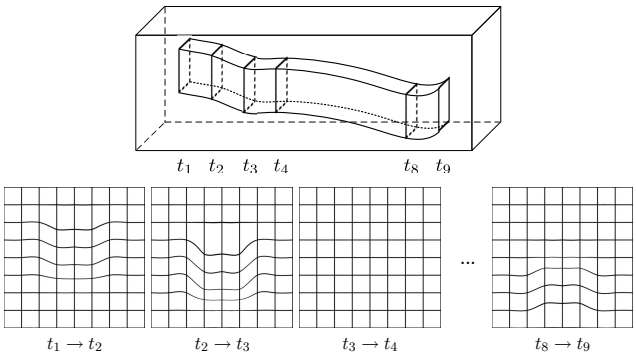

Figure 4. The illustration of the warp obtained by polyharmonic interpolation. It can be observed that the spatio-temporal information of a moving object is propagated for a long period.

4. Spatio-temporal Video Re-localization

Given a query video and a reference video, STVR aims to localize tubelets in the reference video such that the tubelets semantically correspond to the query video. To achieve the goal, we design a novel model detecting bounding boxes in the reference video based on the matching results between the query and reference videos. Our proposed model is shown in Figure 5.

4.1. Video Feature Extraction

For the STVR task, both the temporal and spatio information should be captured in the raw video feature. Hence, we choose inflated 3D ConvNet (I3D) [4] as the feature extractor. The I3D model is originally trained on 64-frame video snippets and tested on 250-frame video snippets. Using many frames together to extract video features is fine for video classification task, but it may not be a good idea for the spatio-temporal localization task because the regions we want to locate may move over a long distance. Therefore, we reduce the number of frames of the video snippets to 8.

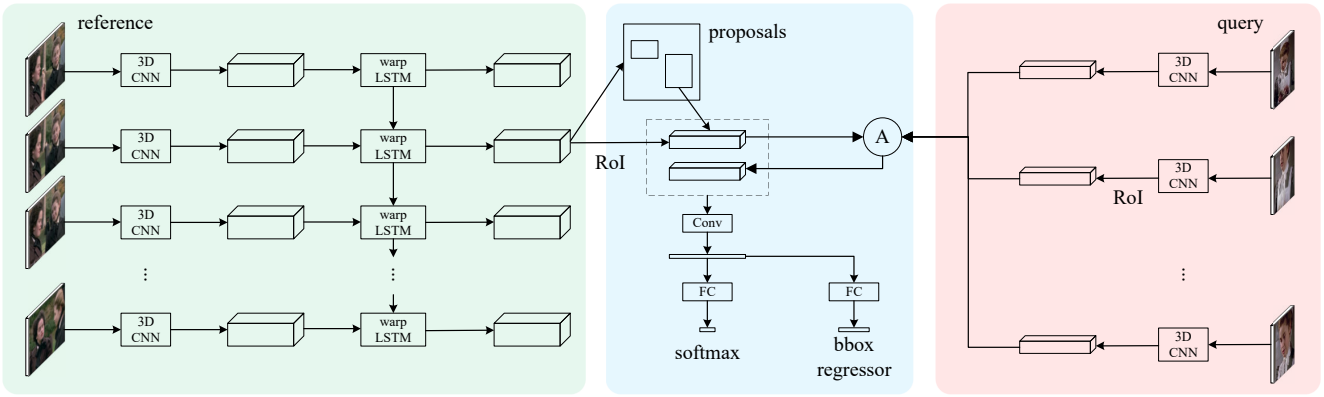

Figure 5. The architecture of our proposed model for STVR. The inputs are a query and a reference video. Both query and reference are split into clips and then fed into a 3D CNN to extract video features. Later, the long-term spatio-temporal information in the reference video is aggregated by the warp LSTM to produce a new reference feature. Region proposal network [38] is applied on the new reference feature to generate several proposals. For each proposal, we use an attention mechanism to select the most related query feature and concatenate the proposal feature with the attention weighted query feature. The concatenated feature is used for the second stage prediction, which outputs a refined bounding box and a binary label indicating whether the query and the proposal are semantically corresponding to each other. Ⓐ denotes the attention mechanism, and the dashed rectangle means concatenating along the channel dimension.

We also re-sample all the videos at the FPS of 24 so that each snippet is just $\frac{1}{3}$ second long. We choose the activation values at the "Mixed_4c" layer in the I3D model as the video feature, which has a spatio stride of 16 and a temporal stride of 4.

Let $r_i \in \mathbb{R}^{8\times H\times W\times 3}$ denote the i-th reference clip, where H and W are the height and width of the reference video, respectively. The feature extraction for the reference is given by:

$$\hat{f}_i^r = \text{I3D}(r_i), \tag{5}$$

where $\hat{f}_i^r \in \mathbb{R}^{2\times \frac{H}{16}\times \frac{W}{16}\times 512}$ is the extracted feature for the i-th reference clip. The 4D feature is transformed to 3D by flattening along the temporal dimension and the channel dimension:

$$f_i^r = \text{flatten}(\hat{f}_i^r), \tag{6}$$

where $f_i^r \in \mathbb{R}^{\frac{H}{16}\times \frac{W}{16}\times 1024}$ is the flattened feature. For the j-th clip in the query video q_j, we apply 2D RoI pooling after 3D convolution to generate a fixed size feature:

$$f_j^q = \text{RoI}\left(\text{flatten}\big(\text{I3D}(q_j)\big)\right), \tag{7}$$

where $f_j^q \in \mathbb{R}^{7\times 7\times 1024}$ is the j-th query feature.

4.2. Reference Propagation

The extracted f_i^r only contains the spatio-temporal information within the 8-frame clip. To propagate the spatio-temporal information from previous clips of the reference video to the i-th clip for better re-localization, we add a warp LSTM layer to update the reference feature.

$$h_i = \text{warpLSTM}(f_i^r, h_{i-1}), \tag{8}$$

where $h_i \in \mathbb{R}^{\frac{H}{16}\times \frac{W}{16}\times 1024}$ is the hidden state of the warp LSTM, which also serves as a new reference representation.

4.3. Proposal Generation

The proposal generation module aims to find all the bounding boxes containing the content of potential interest in one clip. The generation of reference proposals is designed following Faster RCNN [38]. h_i is fed into the region proposal network (RPN) to generate proposals:

$$\begin{aligned} p_k &= \text{RPN}(h_i), \\ f_k^p &= \text{RoI}(h_i, p_k), \end{aligned} \tag{9}$$

where p_k is the predicted bounding box for the k-th proposal and $f_k^p \in \mathbb{R}^{7\times 7\times 1024}$ is the feature of the k-th proposal obtained by RoI pooling.

4.4. Query and Reference Matching

We match every proposal in the reference clip with the query video. The query video may be much longer than one clip in the reference, which has only 8 frames. As such, some parts in the query video may not well correspond to a short proposal. Motivated by [13, 50], we design an attention mechanism to select which part in the query video should be matched with the proposal. For the k-th proposal, the features of the query video are attentively summarized as:

$$\begin{aligned} e_{k,j} &= \tanh(W^q \text{avg}(f_j^q) + W^r \text{avg}(f_k^p) + b_p), \\ \alpha_{k,j} &= \frac{\exp(w^\top e_{k,j} + b_s)}{\sum_i \exp(w^\top e_{k,i} + b_s)}, \\ \bar{f}_k^q &= \sum_j \alpha_{k,j} f_j^q, \end{aligned} \tag{10}$$

where $\bar{f}_k^q$ is the weighted query representation. W^q, W^r, w are the weight parameters in the attention model with b_p and

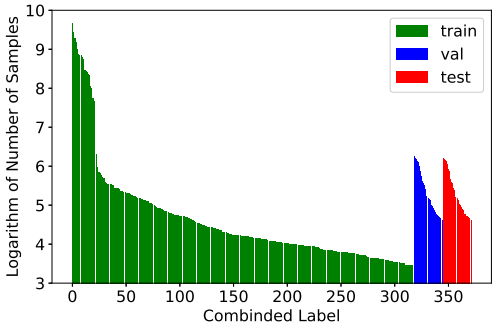

Figure 6. The distribution of the number of samples for each combined label. The combined labels belonging to training, validation, and testing sets are marked by green, blue, and red, respectively. The combined labels with less than 32 tubelet samples in the training set are omitted for clarity.

b_s denoting the bias terms. avg$(\cdot)$ means average pooling along the spatio dimensions.

4.5. Label and Bounding Box Predictions

The proposal feature f_k^p and the attentively weighted query feature $\bar{f}_k^q$ are concatenated along the channel dimension for the final label prediction and bounding box refinement. The final label is binary, indicating whether the query video and the k-th proposal are semantically corresponding to each. The ground-truth label will be "true" if the query video and the k-th proposal are indeed semantically corresponding to each other. Otherwise, the ground-truth label will be "false". The bounding box regression layers are designed following [38]. Please refer to [38] for more details.

5. The Reorganized Datasets

Existing video datasets are designed for other vision tasks, such as classification [35], temporal localization [1], action recognition [43], captioning [6], and video summarization [17]. None of them is suitable for the STVR task, which requires pairs of query and reference videos. The query should semantically correspond to some labeled tubelets in the reference video. It will require a huge expensive labor to collect and annotate such a video dataset.

As such, we propose to reorganize the AVA dataset for the STVR task. The AVA dataset [16] is originally designed for the spatio-temporal action localization task. There are 430 15-minute video clips with per second action bounding box annotations. The annotated actions are 80 categories of atomic actions, including "stand", "watch", "listen", *etc.* The actions are exhaustively annotated, which results in 1.58 million action annotations with multiple labels per person. The first step of the reorganization is to generate tubelets by linking the labeled bounding boxes at each second. We will link two bounding boxes if they are the consecutive bounding boxes of the same subject with all the action labels being the same. After linking, the tubelets with exactly the same action labels are regarded as semantically corresponding to each other. For example, a tubelet labeled with "stand + talk to" semantically corresponds to other tubelets labeled with "stand + talk to" as well. The tubelet does not correspond to the tubelets labeled with "stand" only, "talk to" only, or "sit + talk to". It can be understood as that the multiple atomic action labels annotated with one bounding box are combined together.

Different from spatio-temporal action detection, STVR aims to semantically match video tubelets beyond a predefined category list. Thus, we further split the video tubelets according to their combined labels following [13], so that the training categories have no overlap with the validation or testing categories. We first choose 54 combined labels having over 100 tubelet samples from the 64 validation videos. 27 of them are used for validation and the other 27 are used for testing. After fixing the 54 combined labels, we remove all the frames overlapping with the tubelets belonging to the 54 combined labels in the 235 training videos. The left tubelets in the 235 training videos are used to train our STVR model. The numbers of tubelets belonging to different combined labels are shown in Figure 6.

We describe how to create the query and reference pairs in the following. The combined action labels having only one tubelet sample are all discarded because no pair can be formed for this combined label. For any query tubelet, we randomly find another tubelet having the same combined label as the target. Then we crop the whole segment containing the target tubelet as the reference video. Such cropping will simplify the STVR task because the temporal boundary is known. To avoid this, we crop a segment longer than the target tubelet so that the reference video contains some background before and after the target tubelet. One thing to mention is that the cropped reference video may contain more than one tubelet having the same label as the query. All of the tubelets in the reference sharing the same label as the query are regarded as target tubelets. During training, the query tubelet and reference video are randomly paired, while the pairs are fixed for validation and testing.

Following the same intuition, we also reorganize the videos in the UCF-101-24 dataset [43] for experiments. Among the 24 action categories, 14, 5, and 5 classes are used for training, validation, and testing, respectively.

6. Experiments

We conduct several experiments to verify the effectiveness of warp LSTM in solving the STVR problem. First, three baseline methods are designed and introduced. Then we introduce our experimental settings including evaluation criteria and implementation details. Finally, we report the quantitative results and show the visualizations.

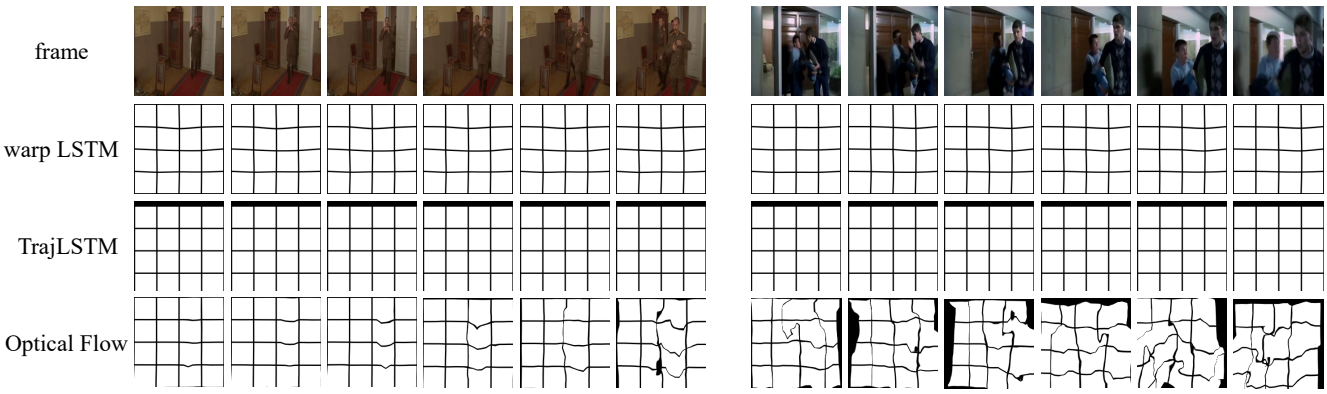

Figure 7. The visualization of the warped grids with different methods. Only one of the five links in TrajLSTM is shown here.

6.1. Baseline Methods

Existing spatio-temporal localization methods mainly focus on localizing objects or actions in videos. As far as we know, there is no method specifically designed for STVR. So we design three baseline models for comparison.

Clip Independent Baseline. Clip independent baseline is designed based on the spatio-temporal action localization methods [22, 31]. The reference video is divided into a series of 8-frame clips and the bounding box prediction only depends on the information within the current clip. The clip independent baseline can be implemented by just removing the warp LSTM layer in our proposed model in Figure 5.

Other ConvLSTM Variants. The proposed warp LSTM can be viewed as a variant to ConvLSTM [39]. So we create a baseline to compare with the original ConvLSTM by replacing warp LSTM with ConvLSTM. Similarly, we also create a baseline for the comparison with TrajGRU [40]. We replace the polyharmonic interpolation with the structure generating network in [40] and name this baseline as TrajLSTM.

Optical Flow Baseline. Warping images by optical flow has been widely used in computer vision research. It is also possible to warp the hidden state of ConvLSTM by the accumulated optical flow. We create another baseline in which the hidden state of ConvLSTM is warped according to optical flow.

6.2. Experimental Settings

We resize all the videos to the resolution 320×320 before feeding them into the CNN models. The I3D model we use is first initialized by training on the Kinetics dataset [4] and then fine-tuned during the training of our model. To form a batch during the training process, the length of the reference video needs to be fixed. The reference video is fixed to be 2 seconds long by randomly cropping or padding zeros. During testing, the query and reference video in full length are fed into the model without batching. For warp LSTM, we put three horizontal and three vertical lines on the 20×20 feature map, which leads to nine control points: $\{(5,5), (5,10), (5,15), (10,5), (10,10), (10,15), (15,5), (15,10), (15,15)\}$. The displacements of the control points are predicted by one CNN layer with a kernel size of 1×1. To reduce the number of model parameters, the input-to-state and state-to-state convolutions in warp LSTM are designed following the bottleneck block [18]. The 1024-channel feature map is first projected to 128-channel and a skip connection is also added from the input to the output of warp LSTM. The following region proposal layers and bounding box regression layers are implemented by Tensorflow Object Detection API [24]. The number of links for TrajLSTM is set to be 5. FlowNet2 [25] is used for optical flow extraction in the optical flow baseline. The optical flow is resized and rescaled by a factor of $\frac{1}{16}$ to fit the size of the feature map.

All the models are trained using stochastic gradient descent (SGD) with momentum value 0.9. The initial learning rate is 0.03 and is divided by 10 after 10k iterations. The batch size we use is set to be 8. It takes about five hours to train one model on four Tesla P40 until the convergence.

6.3. Evaluation Metrics

The frame-mAP computed using a modified version of the code[6] released by official the AVA dataset website is reported for evaluation. As described in Sec. 5, there are 27 combined labels for testing. Given a pair of query and reference video, all the labeled bounding boxes in the reference belonging to the same combined label with the query are regarded as ground-truth. The bounding boxes predicted with positive labels are regarded as predictions. The average precision (AP) for one combined label is computed over all the ground-truths and predictions belonging to that combined label with IoU over 0.5. We report the mAP, which is the average of the AP values over the 27 testing combined labels.

[6] https://github.com/activitynet/ActivityNet/blob/master/Evaluation/get_ava_performance.py

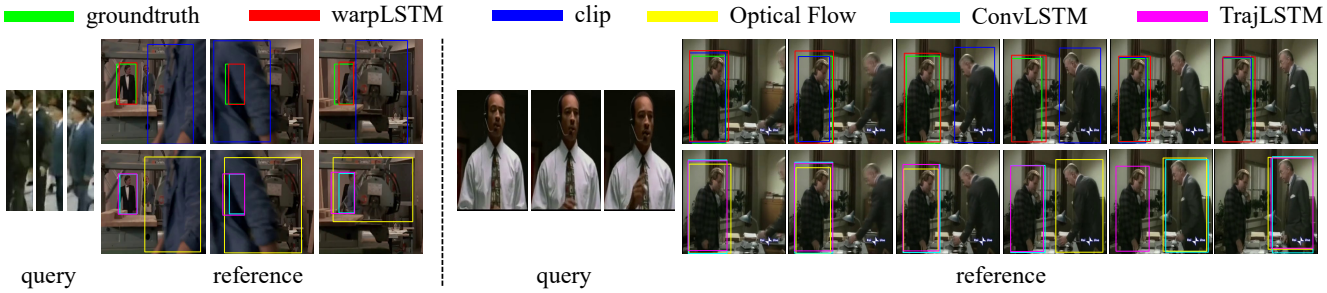

Figure 8. The visualization of the re-localization results. The bounding boxes with the largest confidence of different methods are shown in different colors.

Table 1. The frame-mAP computed with IoU threshold 0.5 of all the methods.

Method	AVA	UCF-101-24
Clip	18.8	52.9
ConvLSTM [39]	20.8	53.5
Optical Flow	20.2	52.0
TrajLSTM [40]	21.0	54.8
Warp LSTM	21.8	59.4

6.4. Quantitative Results

The quantitative results of all the methods are shown in Table 1. Meanwhile, Figure 7 shows the warp visualization of two videos in the test split. By comparing the mAP of "Clip" and "'ConvLSTM' baseline, we find that propagating the spatio-temporal information at previous time-steps to current time-step is better than doing the prediction independently for each clip. Using accumulated optical flow to warp the hidden state of the previous time-step leads to worse results than ConvLSTM, which may be because the error is too large in the accumulated optical flow. It can be seen in Figure 7 that the girds warped by optical flow are noisy. The mAP values of TrajLSTM and ConvLSTM are very similar. The links learned by TrajLSTM on complex action videos seem to be some fixed offsets. The performance of the warp LSTM is the best of all the methods. The results show that propagating the long-term spatio-temporal information by warp LSTM is helpful to STVR.

6.5. Qualitative Results

In the second row in Figure 7, it can be observed that warp LSTM is able to detect the moving actor and warp the previous feature maps to compensate for the movement. The black in the third and fourth row means that these two methods try to warp some regions outside the feature map into the outputs. Figure 8 is the visualization of two STVR results. The combined label of the first and second queries are "walk + talk to + watch" and "stand + answer phone", respectively. In the second clip of the first reference video, the person in the ground-truth bounding box is totally occluded. "Clip" and "Optical flow" baseline fail to localize correctly because of the occlusion. However, the other three methods are able to handle the short occlusion because they can use the spatio-temporal information in previous clips. In the second example, the two men in the reference video are both standing. The man on the left is labeled with "stand + answer phone" and the man on the right is labeled with "stand + touch + listen to". It is difficult to distinguish the combined label of these two men because their actions look similar. "Clip", "ConvLSTM", "Optical Flow" and "TrajLSTM" make at least one error among the six clips, while the proposed warp LSTM correctly localizes the left man all the time.

7. Conclusion

In this paper, we tackled the spatio-temporal video re-localization problem for the first time. Given a query video, spatio-temporal video re-localization aims to find tubelets in a reference video such that the tubelets are semantically corresponding to the given query. Spatio-temporal video re-localization is a natural extension of the temporal video-relocalization [13] which can be applied to video retrieval and surveillance. To make spatio-temporal video re-localization research possible, we created a new dataset by reorganizing the videos in the AVA dataset [16]. Furthermore, we proposed a matching model to capture the semantic relationship between the query and reference videos. The long-term spatio-temporal information is propagated by a warp LSTM to generate better bounding box predictions. The extensive experimental results show that our proposed method is superior to baseline methods on the spatio-temporal video re-localization task.

In the future, we plan to integrate the warp operation into more sophisticated models such as [15, 45, 57].

Acknowledgement

This work is partially supported by NSF awards 1704309, 1722847, and 1813709.

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[2] Liujuan Cao, Rongrong Ji, Yue Gao, Wei Liu, and Qi Tian. Mining spatiotemporal video patterns towards robust action retrieval. *Neurocomputing*, 2013.

[3] Liujuan Cao, Xian-Ming Liu, Wei Liu, Rongrong Ji, and Thomas Huang. Localizing web videos using social images. *Information Sciences*, 2015.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[5] Shih-Fu Chang, William Chen, Horace J Meng, Hari Sundaram, and Di Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *CSVT*, 1998.

[6] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[7] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018.

[8] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *AAAI*, 2019.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[10] Xinpeng Chen, Lin Ma, Wenhao Jiang, Jian Yao, and Wei Liu. Regularizing rnns for caption generation by reconstructing the past with the present. In *CVPR*, 2018.

[11] Xin Chen, Chengcui Zhang, Shu-Ching Chen, and Stuart Rubin. A human-centered multiple instance learning framework for semantic video retrieval. *IEEE Transactions on Systems, Man, and Cybernetics*, 2009.

[12] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *CVPR*, 2017.

[13] Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. Video re-localization. In *ECCV*, 2018.

[14] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017.

[15] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018.

[16] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[17] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[19] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[21] Anthony Hoogs, AG Amitha Perera, Roderic Collins, Arslan Basharat, Keith Fieldhouse, Chuck Atkins, Linus Sherrill, Benjamin Boeckel, Russell Blue, Matthew Woehlke, et al. An end-to-end system for content-based video retrieval using behavior, actions, and appearance with interactive query refinement. In *AVSS*, 2015.

[22] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, 2017.

[23] Jun-Wei Hsieh, Shang-Li Yu, and Yung-Sheng Chen. Motion-based video retrieval by trajectory matching. *CSVT*, 2006.

[24] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017.

[25] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[26] Armin Iske. *Multiresolution methods in scattered data modelling*. Springer Science & Business Media, 2004.

[27] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2013.

[28] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *ECCV*, 2018.

[29] Yu-Gang Jiang, Qi Dai, Wei Liu, Xiangyang Xue, and Chong-Wah Ngo. Human action recognition in unconstrained videos by explicit motion modeling. *TIP*, 2015.

[30] Yu-Gang Jiang, Jiajun Wang, Qiang Wang, Wei Liu, and Chong-Wah Ngo. Hierarchical visualization of video search results for topic-based browsing. *TMM*, 2016.

[31] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[32] Atsushi Kanehira, Luc Van Gool, Yoshitaka Ushiku, and Tatsuya Harada. aware video summarization. In *CVPR*, 2018.

[33] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *CVPR*, 2017.

[34] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[35] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola,

Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[37] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *ECCV*, 2018.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[39] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[40] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NIPS*, 2017.

[41] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavier Giro-i Nieto, and Shih-Fu Chang. Online detection of action start in untrimmed, streaming videos. In *ECCV*, 2018.

[42] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[43] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[44] Chih-Wen Su, Hong-Yuan Mark Liao, Hsiao-Rong Tyan, Chia-Wen Lin, Duan-Yu Chen, and Kuo-Chin Fan. Motion flow-based video retrieval. *TMM*, 2007.

[45] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018.

[46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[47] Rene Visser, Nicu Sebe, and Erwin Bakker. Object recognition for video retrieval. In *International Conference on Image and Video Retrieval*, 2002.

[48] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[49] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.

[50] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.

[51] Rong Yan, Alexander G Hauptmann, and Rong Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *MM*, 2003.

[52] Hongtao Yang, Xuming He, and Fatih Porikli. One-shot action localization by learning sequence matching network. In *CVPR*, 2018.

[53] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[54] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[55] Kaihao Zhang, Yongzhen Huang, Yong Du, and Liang Wang. Facial expression recognition based on deep evolutional spatial-temporal networks. *TIP*, 2017.

[56] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *TIP*, 2019.

[57] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state lstm for text representation. In *ACL*, 2018.

[58] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *CVPR*, 2018.