Spatio-temporal Video Re-localization by Warp LSTM

Yang Feng$^2*$ Lin Ma$^2$† Wei Liu$^2$ Jiebo Luo$^2$

$^2$Tencent AI Lab $^3$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.

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-
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 is concatenated with the current input for further computation. Different from ConvLSTM, the previous hidden state can 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 3D 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-
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:

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

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 \(W_s\) and \(b_s\) are the parameters of the ConvLSTM layer. \(*\) is the convolution operation and \(\odot\) is the element-wise product. \(\sigma\) and \(\phi\) are sigmoid non-linearity and hyperbolic tangent non-linearity, 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.

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 \(p_t\). Please note that the content in \(q_{t-1}\) may be not semantically related to the object in \(p_t\).

![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.](image)

![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 \(p_t\).](image)
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]:

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

where \( \phi_k \) 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:

\[
\begin{align*}
    d_{t-1} &= W_{xd} * x_t + W_{hd} * h_{t-1} + b_d, \\
    h'_{t-1} &= \text{warp}(h_{t-1}, d_{t-1}), \\
    c'_{t-1} &= \text{warp}(c_{t-1}, d_{t-1}), \\
    i_t &= \sigma(W_{xI} * x_t + W_{hI} * h'_{t-1} + b_i), \\
    g_t &= \sigma(W_{xG} * x_t + W_{hG} * h'_{t-1} + b_g), \\
    f_t &= \sigma(W_{xf} * x_t + W_{hf} * h'_{t-1} + b_f), \\
    a_t &= \sigma(W_{xa} * x_t + W_{ha} * h'_{t-1} + b_a), \\
    c_t &= f_t \odot c'_{t-1} + i_t \odot g_t, \\
    h_t &= a_t \odot c_t,
\end{align*}
\]

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. \( \text{warp}(\cdot, \cdot) \) is the sparse image warping function\(^3\), 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 \( 3 \times 3 \) 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.

### 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.
We also re-sample all the videos at the FPS of 24 so that each snippet is just $\frac{1}{4}$ 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:

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

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:

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

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:

$$f_j^q = \text{RoI}(\text{flatten}(\text{I3D}(q_j))),$$

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.

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

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 R-CNN [38]. $h_i$ is fed into the region proposal network (RPN) to generate proposals:

$$p_k = \text{RPN}(h_i),$$

$$f_k^p = \text{RoI}(h_i, p_k),$$

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:

$$e_{k,j} = \tanh(W^q \text{avg}(f_j^q) + W^r \text{avg}(f_k^p) + b_p),$$

$$\alpha_{k,j} = \frac{\exp(w^\top e_{k,j} + b_s)}{\sum_i \exp(w^\top e_{k,i} + b_s)},$$

$$\bar{f}_k^q = \sum_j \alpha_{k,j} f_j^q,$$

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
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.
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 $320 \times 320$ before feeding them into the CNN models. The 13D 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 $20 \times 20$ 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 $1 \times 1$. 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
Table 1. The frame-mAP computed with IoU threshold 0.5 of all the methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>AVA</th>
<th>UCF-101-24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>18.8</td>
<td>52.9</td>
</tr>
<tr>
<td>ConvLSTM [39]</td>
<td>20.8</td>
<td>53.5</td>
</tr>
<tr>
<td>Optical Flow</td>
<td>20.2</td>
<td>52.0</td>
</tr>
<tr>
<td>TrajLSTM [40]</td>
<td>21.0</td>
<td>54.8</td>
</tr>
<tr>
<td>Warp LSTM</td>
<td>21.8</td>
<td>59.4</td>
</tr>
</tbody>
</table>

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 grids 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

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


