TORNADO: A Spatio-Temporal Convolutional Regression Network for Video Action Proposal

Hongyuan Zhu*
I^{2}R, A*Star, Singapore
zhuh@i2r.a-star.edu.sg

Romain Vial*
MINES ParisTech, France
romain.vial@mines-paristech.fr

Shijian Lu
NTU, Singapore
Shijian.Lu@ntu.edu.sg

Abstract

Given a video clip, action proposal aims to quickly generate a number of spatio-temporal tubes that enclose candidate human activities. Recently, the regression-based networks and long-term recurrent convolutional network (LRCN) have demonstrated superior performance in object detection and action recognition. However, the regression-based detectors perform inference without considering the temporal context among neighboring frames, and the LRCN using global visual percepts lacks the capability to capture local temporal dynamics. In this paper, we present a novel framework called TORNADO for human action proposal detection in un-trimmed video clips. Specifically, we propose a spatio-temporal convolutional network that combines the advantages of regression-based detector and LRCN by empowering Convolutional LSTM with regression capability. Our approach consists of a temporal convolutional regression network (T-CRN) and a spatial regression network (S-CRN) which are trained end-to-end on both RGB and optical flow streams. They fuse appearance, motion and temporal contexts to regress the bounding boxes of candidate human actions simultaneously in 28 FPS. The action proposals are constructed by solving dynamic programming with peak trimming of the generated action boxes. Extensive experiments on the challenging UCF-101 and UCF-Sports datasets show that our method achieves superior performance as compared with the state-of-the-arts.

1. Introduction

Human can understand a video by analyzing how many peoples are inside, where are they and how they interact with each other in a single glance. Developing a fast and accurate algorithm which can well mimic this human capability can unlock great potentials for a series of responsive and automatic applications, such as assistive robots and video surveillance [7, 43, 24, 20]. However, directly feeding raw video volumes to existing learning algorithms is computationally extensive. On the other hand, human visual system assigns computational resources to a small number of regions that correspond to salient and meaningful human actions, which highly motivates the development of video action proposal.

Action proposal takes an un-trimmed video as input and outputs a number of tubes which consist of the potential human action boxes. It is a challenging task due to the illumination variation, abrupt motion changes, background clutters, etc., and so requires a systematic and principled modeling approach. In contrast to static object proposal [4, 44, 29] which considers appearance cue only, action proposal in videos is much more complex and requires to integrate several components including (1) a fast and accurate candidate action box generator that can simultaneously consider appearance, motion and temporal context; (2) a robust and accurate actionness estimator for the generated proposals; (3) an efficient association mechanism to link the candidate boxes into action proposals.

With the above objectives in mind, a number of action proposal methods [11, 22, 39, 42, 16, 9] have been developed in recent years and very promising results have been achieved. Different approaches have been explored including segmentation-and-merging [11, 22, 19], dense trajectories clustering [39], human detection [42], deep learning [16], etc. However, segmentation-and-merging [11, 22, 39, 26] and human detection [42] approaches rely on low-level cues which fall behind the state-of-the-art. Recently, Li *et al.*[16] explored to use Region Proposal Network (RPN) [29] for action box detection, but RPN processes each video frame individually and ignores the temporal context among adjacent video frames. They trained additional detector using low-level descriptors to remedy detection misses, but such separate sequential processing approach accumulates error and is also difficult to optimize.

To handle these challenges, we propose a new framework **TORNADO** for video action proposal. The TORNADO consists of a temporal convolutional regression network (T-CRN) and a spatial convolutional regression network (S-

*H.Zhu and R.Vial are first authors with equal contributions.

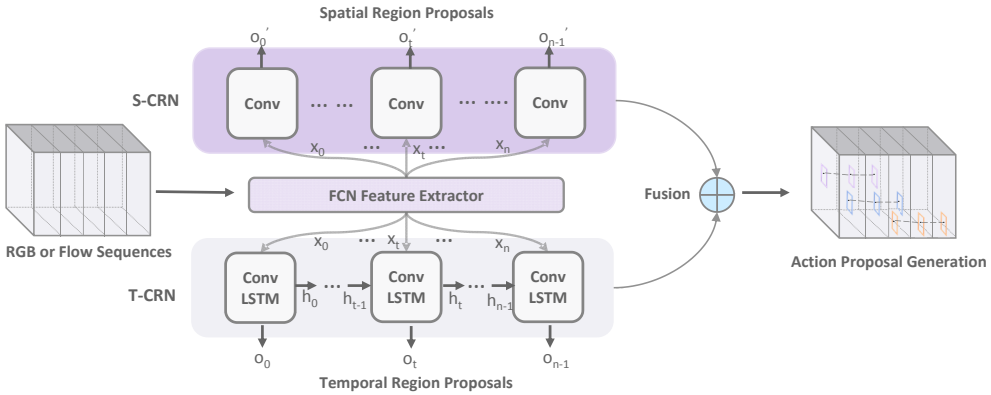

Figure 1. Conceptual illustration of our method. Our method consists of a spatial convolutional regression network (S-CRN) and a temporal convolutional regression network (T-CRN). Both neworks use fully convolutional networks (FCN) to extract discriminative deep features $x_t, t \in \{0, n\}$ from RGB or Flow frames. Then the T-CRN performs sequential inference using ConvLSTM (Sec.3.1) to regress temporal bounding boxes (or region proposals) $o_t, t \in \{0, n\}$ of candidate human actions, and the S-CRN regresses spatial candidate action boxes $o'_t, t \in \{0, n\}$ using cues in single frame with normal convolutional layer. These candidate boxes are then fused and the action proposals are finally constructed by using efficient dynamic programming and peak pattern trimming.

CRN) which are combined to regress the bounding boxes of human action proposals. The T-CRN is inspired by recent advances in regression-based image detector [28, 18, 29] and LRCN [5] for action classification. Specifically, we replace the normal LSTM in LRCN with a Convolutional LSTM (ConvLSTM) to preserve the spatio-temporal smoothness among neighboring pixels and frames. In addition, we exploit temporal context with recurrent unfolding to empower ConvLSTM with regression capability (for candidate boxes prediction) which is largely neglected in the recent literature. The S-CRN shares a similar architecture as T-CRN which fully exploits the appearance cues in a single frame to improve the action candidate boxes generation performance. Action proposals are finally constructed by solving dynamic programming with peak-trimming. Extensive experiments on both UCF-101 and UCF-Sports datasets demonstrate that the proposed TORNADO achieves superior performance as compared with the state-of-the-arts.

Our contributions are two folds. First, we propose the first framework that integrates the complementary spatial and temporal information into an end-to-end trainable system for video action proposal with state-of-the-art performance. Second, a novel and efficient path trimming method is proposed to handle untrimmed video by examining actioness and background score pattern without using extra detectors.

2. Related Work

Recurrent Neural Network: Recurrent Neural Networks, especially LSTM [10] have become recent prevalent recurrent structure because it incorporates memory that has explicit control of when to 'forget' and when to 'update' given new information. LSTM has been used for video action classification [5], image-sentence generation [12], object detection refinement [21] etc. Although LSTM is suitable for certain time series tasks, it treats all input data as flatten vectors which ignores the spatial correlation among adjacent pixels. Hence they are more focus on global appearance changes as argued by Ballas *et al.* [1].

The Convolutional LSTM [33, 1] has recently become popular because it has a smaller memory footprint by using convolution layer with a sparser connection instead of the fully dense connection in LSTM, hence should theoretically better to capture the spatio-temporal smoothness change among local pixels in video related tasks. Actually, ConvLSTM has been applied to image segmentation [30], optical flow estimation[23] and video action classification [17] with promising performance. However, the study to extend ConvLSTM for sequence bounding boxes regression task is largely neglected in recent literature to the best of our knowledge. Our experiment shows that ConvLSTM with regression capability is effective for video action proposal.

Video based Action Proposal: Early success in action detection is based on exhaustive search using sliding cuboids [13, 15, 35]. The rigid cuboid often lacks sufficient capacity to capture the versatile shape of the human actions. Besides cuboid search, structure output regression [37, 36] has been explored to detect spatial-temporal action tubes but it can only search the best action path with fixed size window. Recently, Yu *et al.*[42] address the limitation of the fixed detection window by replacing it with human detectors and use max sub-path search for action proposal. Several recent systems[39, 3, 2] generate action proposals by clustering long term point trajectories, but they cannot detect actions with small motions

well [39]. Some segmentation-and-merging strategy is proposed in [11, 22, 19] where videos are firstly segmented and then hierarchically merged into action proposals. On the other hand, the quality of video segmentation becomes the bottleneck especially under unconstrained setting. Furthermore, aforementioned methods are based on low-level cues, which are sensitive to abrupt appearance and scale changes and accordingly degrade their performance greatly.

Recently, deep learning based on frame-level region proposal have become a new paradigm for spatial temporal action localization. Shou *et al.* [34] propose to use CNN for action localization in temporal domain. Gkioxari and Malik[9] uses appearance and motion based two-stream R-CNN[8] with SelectiveSearch[38] to find more accurate spatial-temporal candidate action regions which are further seamed together using dynamic programming. However, it needs to search the proposal region class-by-class and their method can only work on temporally trimmed video. Saha *et al.*[32] and Peng *et al.*[25] extends the approach in [9] by using Fast-RCNN which first detect class-specific action regions, and then seam them to be class-specific action tubes using dynamic programming. Vial *et al.* [40] propose to use LSTM based regression detector, however, they are based on dense connection, which can only runs at 5FPS. Our system also uses dynamic programming to seam action proposal, but we propose a novel path trimming approach by checking the score pattern along the path for better efficiency and accuracy in untrimmed video.

Detection-and-tracking methods have also been used for action localization and action proposal. For example, Weinzaepfel *et al.*[41] train a two-stream R-CNN to detect class-specific actions one-by-one for tracking. Later, Li *et al.*[16] use a single stream Faster-RCNN [29] to replace the R-CNN which achieves the state-of-the-art performance. An improved dynamic programming of [42] is designed to generate action proposal where the missing detection are remedied by tracking-by-detection.

Our method belongs to the two-stream deep learning based approach. We train a S-CRN detector to detect frame-level action candidates in the similar way. In addition, we perform the reasoning by training a T-CRN detector to capture the long-term dependency and contexts among adjacent frames which are largely neglected in most existing methods. Furthermore, we approximate detection-by-tracking methods using efficient dynamic programming and trimming. Experiments demonstrate superior performance of our proposed method as compared with the state-of-the-arts using extra detectors.

3. Methodology

The core idea of our work is to formulate the video action proposal task as a sequence bounding boxes regression problem using RNN. It first maps a variable-length untrimmed video into sequences of candidate action boxes. Action proposals are then constructed by linking up these boxes by using an efficient dynamic programming and trimming technique. Fig. 1 shows an overview of our method.

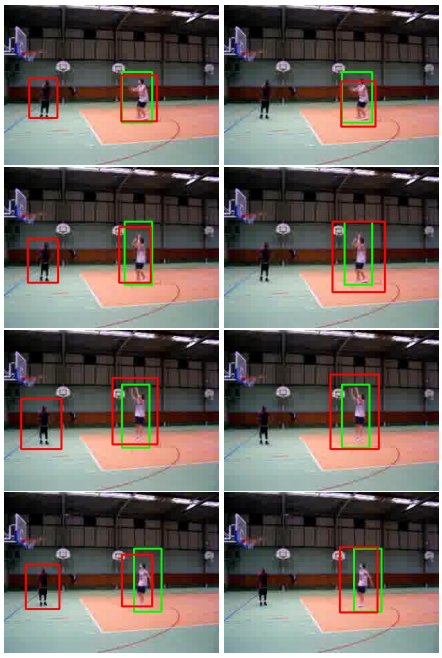

Figure 2. Human action not only accounts for local kinematics and motions, the temporal context of adjacent frames is also important. The S-CRN detector (left column) performs inference using single image cues and it produces false alarms for the player dressed in black, while the T-CRN detector (right column) can resolve such ambiguity by using information from earlier frames. The green boxes are the ground-truth, while the red-boxes are the detected action paths with the highest actioness score.

3.1. TORNADO for Action Proposal

Our system consists of a temporal convolutional regression network (T-CRN) and a spatial convolutional regression network (S-CRN) as illustrated in Fig. 1. The T-CRN is mainly responsible for modeling the rich temporal dynamics and contexts, while the S-CRN is mainly used for modeling the rich spatial appearance cues in each frame. We'll first describe T-CRN in the following parts, as S-CRN is a static variant of it with small modification.

Previous deep learning based action proposal methods[16, 9, 41] process video frames individually without using temporal contexts, which often lead to many false-alarm detections while inferring using local patches. Inspired by the recent popular LRCN for video action classification [5], ConvLSTM for image instance segmentation [30] and regression based image object detection [27, 28, 29, 18], the T-CRN explores to empower LRCN with spatio-temporal regression capability for more accurate action proposal generation through examining the

spatio-temporal contexts among neighboring video frames. Fig.2 illustrates the difference between static modeling and temporal modeling.

Fig.1 depicts the structure of the T-CRN. The inference process starts by passing an input frame $f_t \in \mathbb{R}^{h\times w\times c}$ at time t to a fully convolutional neural network, which consists of a series of convolution and pooling layers, to output a spatial-preserved inner representation of the image $x_t \in \mathbb{R}^{h'\times w'\times d}$. This RNN uses a function f_W to map x_t and previous hidden state h_{t-1} into a new hidden state h_t and output o_t. The inference is conducted from left to right, where $h_1 = f_W(x_1, h_0)$, then $h_2 = f_W(x_2, h_1)$, etc, and the h_0 is initialized to zero. During training, an output label l_t will be provided to measure the quality of the prediction o_t which helps to learn the network parameters.

Conventional RNN (including the recent popular LSTM) flattens input feature maps x_t into vectors, where both input-to-hidden and hidden-to-hidden transformations are fully connected. It ignores the spatial correlations among local feature maps which could often leads to sub-optimal results for video activity analysis. The reason is that human actions typically take up a small portion of videos and it is difficult to ensure shift, scale and distortion invariance by using fully connected structures as argued in [1, 17].

The Convolutional LSTM (ConvLSTM) overcomes these limitations by replacing the dense layer of LSTM with the convolutional layers, which preserves the local spatial information and reduce the parameters of the network. The ConvLSTM we use in this work is similarly defined as in [33, 1]:

$$\begin{aligned} i_t &= \sigma(W_{xi} * x_t + U_{hi} * h_{t-1} + \hat{b}_i) \\ f_t &= \sigma(W_{xf} * x_t + U_{hf} * h_{t-1} + \hat{b}_f) \\ o_t &= \sigma(W_{xo} * x_t + U_{ho} * h_{t-1} + \hat{b}_o) \\ g_t &= \sigma(W_{xc} * x_t + U_{hc} * h_{t-1} + \hat{b}_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{1}$$

where $*$ denotes a convolution operation, σ and $\odot$ is the element-wise activation and product function respectively, W_{xi}, W_{xf}, W_{xo}, W_{xc} and U_{xi}, U_{xf}, U_{xo}, U_{xc} are 2D-Convolutional Kernels. $\hat{b}_i, \hat{b}_f, \hat{b}_o, \hat{b}_c$ are the biases. In addition, i_t is the input gate, f_t is the forget gate, o_t is the output gate, g_t is the input modulation gate. c_t is the sum of previous memory cell c_{t-1} which is modulated by forget gate f_t and modulation gate g_t.

The output o_t is a tensor of the shape $S \times S \times (B \times 5 + |C|)$, which means to divide the image into $S \times S$ grids, and each grid predicts B bounding boxes' coordinates and objectness score, hence each bounding box is parametrized by (x, y, w, h, s), where (x, y) represents the center of the box relative to the bounds of the grid cell. The width (height) are normalized with respect to the image's width (height). The confidence s predicts the IOU between the predicted box and any ground-truth box. Moreover, each cell will also predict a pair of action and background score (s_{ac}, s_{bg}), which can be seen as an actionness and a backgroundness score for the given cell, respectively. The label l_t is a tensor of the same shape as o_t, which encodes the ground-truths $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{s}_i, \hat{s}^i_{ac}, \hat{s}^i_{bg})$ for each grid i.

The loss function between l_t and o_t is adapted from YOLO [27] for implementation simplicity:

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \|(x_i, y_i) - (\hat{x}_i, \hat{y}_i)\|^2 \\ &+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \|(\sqrt{h_i}, \sqrt{w_i}) - (\sqrt{\hat{h}_i}, \sqrt{\hat{w}_i})\|^2 \\ &+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (s_i - \hat{s}_i)^2 \\ &+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (s_i - \hat{s}_i)^2 \\ &+ \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c\in\{ac,bg\}} (s_c^i - \hat{s}_c^i)^2 \end{aligned} \tag{2}$$

where 1_i^{obj} denotes if object appears in cell i, 1_{ij}^{obj} denotes that the j^{th} bounding box predictor in cell i is responsible for the prediction (i.e. has the higher IoU with the ground truth between the B boxes) and 1_{ij}^{noobj} denotes that the j^{th} bounding box predictor in cell i is not responsible for the prediction or there is no ground truth boxes in cell i. The notion λ_{coord} and λ_{noobj} are two trade-off factors. The loss function therefore penalizes classification error if object appears in the grid cell. It also penalizes bounding box coordinate error if that predictor has the highest overlap in that grid cell.

T-CRN is doubly deep in spatial-temporal domain which can learn the temporal action dynamics. We also train S-CRN to exploit the rich appearance cues in individual frame for further performance improvement. The S-CRN shares the same architecture as the T-CRN but it replaces the last ConvLSTM layer with a normal Convolutional layer for bounding box regression. Both RGB and Flow stream has an S-CRN and T-CRN, these networks run in parallel. The two sets of bounding boxes from both networks and streams are simply combined into one set. These networks complement each other and fusing their outcomes further improves the performance. The detail architectures of both networks are to elaborated in the implementation details.

3.2. Action Path Linking and Trimming

At the end of the detection process (Sec. 3.1), a set of bounding boxes for each video frame $\mathbf{B} = \{\{b_i^{(j)}, j \in [1 \dots N_{b_i}]\}, i \in [1 \dots T]\}$ are generated, where T is the length of the video and N_{b_i} is the number of predicted boxes in frame i. For each box $b_i^{(j)}$ we have its confidence score $s_c(b_i^{(j)})$, actionness score $s_{ac}(b_i^{(j)})$ and background score $s_{bg}(b_i^{(j)})$. The next objective is to create a set of proposal paths $\mathbf{P} = \{p_i = \{b_{m_i}, b_{m_{i+1}} \dots b_{n_i}\}, i \in [1 \dots |\mathbf{P}|]\}$ where m_i and n_i are the starting and ending frames of the path p_i, respectively.

3.2.1 Action path linking

In order to link frame-level boxes into coherent path, we firstly define a score for a path as in [9] given the confidence scores s_c of each box and the IoU of successive boxes:

$$S(p) = \underbrace{\sum_{i=1}^{T} s_c(b_i)}_{unary} + \lambda_0 \times \underbrace{\sum_{i=2}^{T} IoU(b_i, b_{i-1})}_{pairwise} \quad (3)$$

where λ_0 is a trade-off factor. Maximizing Eq. 3 will give a path whose detection boxes have high confidence scores, and overlap significantly.

Given the set of boxes $\mathbf{B}$, the maximization problem can be solved and the best path $\hat{p}$ found by using the Viterbi algorithm. After the optimal path is found, the boxes in $\hat{p}$ are removed and the algorithm is solved again. This process is repeated until one of the frames contains no predicted boxes.

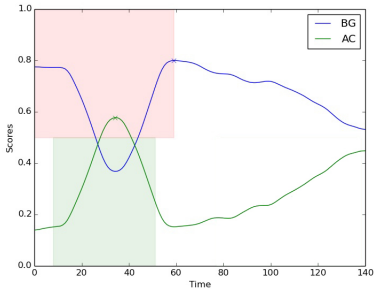

Figure 3. Examples of the peak trimming method on two UCF-101 videos. Blue and green lines are background and actionness scores, respectively. Blue and green crosses are detected peaks in background and actionness scores, respectively. Green patches are ground-truth paths and the red patches are extracted paths with the peak trimming method. One can see that our models are able to trim the initial predicted path accurately thanks to the patterns in the action and background scores.

3.2.2 Trimming with peaks detection

The first pass of dynamic programming aims to extract connected paths by penalizing regions which do not overlap in time. However not all detection boxes within a path exhibit strong actionness scores.

We propose an empirical method by looking at the pattern in the actionness and background scores for the path trimming (e.g. in Fig.3).In particular, the scores are first smoothed by computing their running average. All peaks are then detected for both scores, where a peak is defined as a local maximum among at least n neighbors:

$$\begin{aligned} peaks_{ac} &= \{t, s_{ac}(b_t) = \max(V_n^{(ac)}(t))\} \\ peaks_{bg} &= \{t, s_{bg}(b_t) = \max(V_n^{(bg)}(t))\} \end{aligned} \quad (4)$$

where $V_n^{(k)}(t) = \{s_k(b_i), i \in [t-n \dots t+n]\}, k \in \{ac, bg\}$.

Once we have found the peaks, we can select all subsequences by applying the following algorithm to generate the final action proposals:

$subseq = \emptyset$
for $p \in peaks_{ac}$ **do**
  $s = \max(peaks_{bg} < p)$
  $e = \min(peaks_{bg} > p)$
  add path $\{b_s \dots b_e\}$ at $subseq$
end

The generated proposals are ranked according to their avearaged boxes' scores along the path, the top K (K = 30) paths are kept.

3.3. Implementation Details

The CNN feature extractor we use is based on YOLO [27] which has 24 convolution layers and 2 fully-connected layer. We remove the last two fully-connected layer and replace them with a 1×1 convolution layer with 256 filters. The 1×1 convolution helps to reduce network complexity and improve training stability. The last layer of the T-CRN and S-CRN is a ConvLSTM layer and a convolutional layer with 12 parameters respectively. The input frames are resized to 448×448. The output of T-CRN and S-CRN are of the same shape $7 \times 7 \times 12$, which means to divide the image into 7×7 grids, each grid will predict $S = 2$ actioness scores and $B = 2$ bounding boxes, each of which predicts four coordinates and one objectness score.

We train our network in a two-stream fashion, one is the RGB stream and the other is Flow stream. The flow map is generated by using the dense optical flow [6], where x-, y- and flow magnitude are normalized to the range of [0, 255] to form the final flow map. For the RGB stream, the convolutional part of our model is pre-trained on the ImageNet 1000-class dataset [31]. For the Flow stream, the

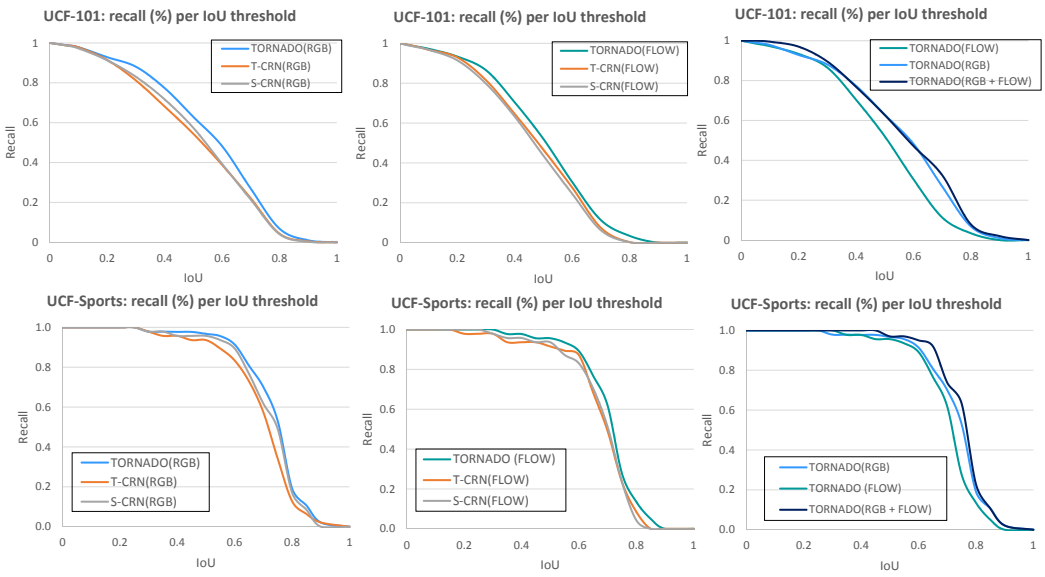

Figure 4. Ablation study of two stream's T-CRN and S-CRN on the UCF-101 (top-row) and UCF-Sports (bottom row) datasets, left column shows RGB stream, middle column shows the Flow stream and right column shows their ensemble.

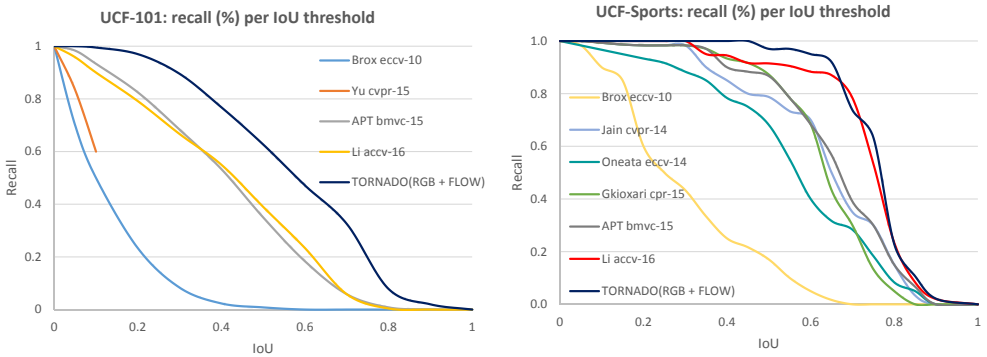

Figure 5. Comparison with other state-of-the-arts on UCF-101 (left) and UCF-Sports (right) datasets, where performance is measured by recall for various IoU thresholds.

convolutional part is pre-trained with the weights of the RGB stream. We make an extensive use of data augmentation (i.e. mirroring, corner and center cropping) to prevent overfitting. We also use Adam [14] optimizer during training with default parameters. While training the S-CRN, a batch size of 32 frames from different videos is set during 100 epochs with an initial learning rate of 10^{-4} decaying at 10^{-5} after the 20^{th} epoch. While the T-CRN, the weighs of the convolutional layers are frozen to avoid a catastrophic forgetting. We then use a batch size of 10 sequences of 10 frames from different videos during 50 epochs. The same learning rate planning is used. We train and test our network on a server with 16 cores XEON, 64G RAM and 4 Nvidia Titan X. Test shows that the T-CRN and S-CRN can run at a 28 FPS.

4. Experiment

In this section, we'll discuss the detail of the experimental evaluation, including the dataset and evaluation metrics, the training details, the ablation study of different components, and the overall performance comparison with baselines and state-of-the-arts and baselines.

4.1. Datasets and Evaluation Metric

We evaluate the proposed approach on two publicly available datasets: UCF-Sports and UCF-101.

UCF-101 dataset has 24 classes containing bounding box annotations of 3,204 videos of which 25% are untrimmed.

UCF-Sports dataset has 150 short sports videos of 10 action classes which has been widely used for action detection. These videos have been trimmed to contain a single action and each frame contains a bounding box annotation

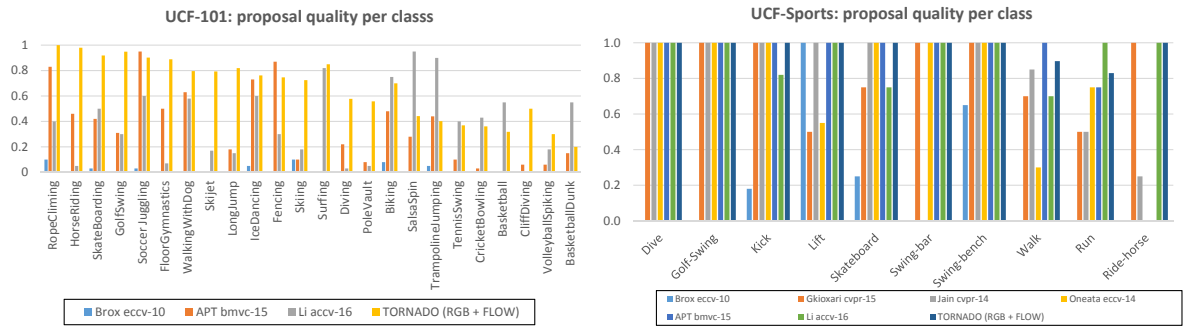

Figure 6. Comparison with other state-of-the-arts on UCF-101 (left) and UCF-Sports (right) datasets, where performance is measured by the recall on each action class.

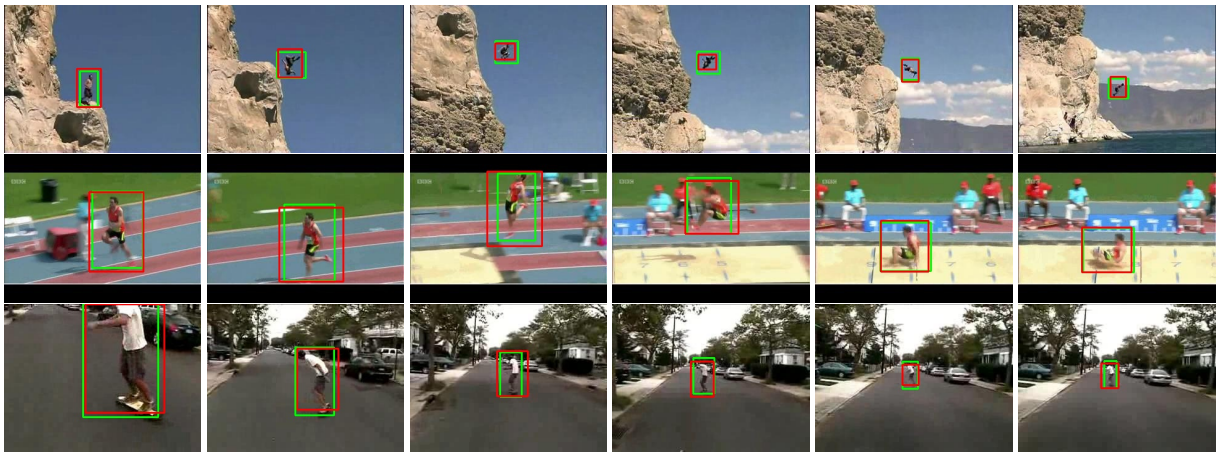

Figure 7. Examples of our method on 3 videos from the UCF101 dataset. Green boxes are ground truth and red boxes are from the best predicted path. Best viewed in colors.

of each action.

Evaluation Metric We follow the metric proposed by van Gemert *et al* [39]. The overlap (OV) between a path $\mathbf{d} = \{d_s \dots d_e\}$ with respect to a ground truth path $\mathbf{g} = \{g_s \dots g_e\}$ is defined as follows:

$$OV(\mathbf{d}, \mathbf{g}) = \frac{1}{|\Theta|} \times \sum_{i \in \Theta} \frac{d_i \bigcap g_i}{d_i \bigcup g_i}$$

where d_s and d_e are the detected bounding boxes in the starting and ending frame of a path, g_s and g_e are the bounding boxes in the starting and ending frame of the ground-truth path, respectively. The Θ denotes the set of frames where there is either **d** and **g** is not empty.

Average Best Overlap **(ABO)** measures the best localization from the set of action proposals $D = \{d_j | j = 1...m\}$ for the each ground-truth annotation G, where Average Best Overlap for a given class c **(ABO(c))** is computed for each ground-truth annotation G_c of class c. The mean ABO **(MABO)** summarizes the performance across all classes. An instance of action, g_i is correctly detected by an action proposal d_j if the overlap score is higher than a threshold η *i.e.*: $OV(d_j, g_i) \geq \eta$ where $\eta \in [0, 1]$. In our work, we target to maximize the recall at a 0.5 threshold as other works [39, 16].

4.2. Ablation Study

We first evaluate the performance of T-CRN, S-CRN and their ensemble TORNADO in RGB, FLOW and RGB+FLOW streams, respectively. Fig. 4 and Tables 1 and 2 shows the ablation analysis of the components.

We first look at the RGB stream. In UCF-101 and UCF-Sports, the S-CRN has a slightly better recall than T-CRN. We think this is because S-CRN captures more static kinetic cues. On the other hand, their ensemble-TORNADO (RGB) improves around 3.5%, which shows the complementarieness between T-CRN and S-CRN where T-CRN is more good at capturing temporal dynamics.

For the FLOW stream, the T-CRN performs slightly better than the S-CRN. This is largely because flow field capture the objects with large dynamic motion, which helps eliminate the interference from the background and allows T-CRN to focus on motion regions, as proved in [17]. On the other hand, the performance of their ensemble also improves around 2.5%, which further confirms the their com-

plementariness.

For the RGB+Flow stream, TORNADO exhibits another 2% improvement for both UCF-101 and UCF-Sports datasets that demonstrates the effectiveness of training on two data modalities. In addition, the performance on the UCF-Sports dataset is significantly better than the UCF-101 dataset, the underlying reason is that UCF-Sports has been trimmed to contain one salient actions only.

4.3. Comparison to state-of-the-arts

We compare our method with state-of-the-arts on both UCF-sports and UCF-101 datasets. We evaluate the recall by varying the IoU threshold (η) in [0, 1]. Fig. 5 shows recall vs. IoU curve for different approaches. In UCF-101 dataset, our approach out-perform [16] by nearly 27%, while the performance of [16] is quite close the unsupervised region merging method [39]. This suggests the validity of complementariness between different modalities and different networks. In UCF-Sports, our method and [16] achieve very high recall in [0, 0.7], which suggest the discriminative of the deep neural networks for action proposal.

Fig. 6 shows the recall performance for each action category. We can observe that our approach out-performs [16] in many classes, especially in the classes which have abrupt motion changes. We conjecture this is more related with our spatial-temporal modeling using RNN.

Finally, we report the overall performance of two datasets in Table. 1 and 2 using several commonly used metrics, including ABO (Average Best Overlap), MABO (Mean ABO over all classes) and average number of proposals per video. Our method achieves the highest MABO and Recall with around 30 proposals. Some qualitative results are displayed in Fig.7, which shows that our method can produce action proposals with good localization. Moreover, Li *et al.*[16] achieves higher ABO than us, but the magnitude between ABO and MABO in [16] is too large which we believe they used a different definition of the metric.

5. Conclusion

We propose a novel framework for searching action proposals in untrimmed video clips. Given a video as input, our method produces a relatively small number of spatially compact and temporally smooth action proposals. The proposed T-CRN detector explored the regression capability of ConvLSTM to sequentially regress the bounding boxes that enclose the action candidates. Together with the S-CRN, our method can exploit the static, motion and temporal context together to produce more robust and accurate bounding boxes within each frame in fast speed. Then these bounding boxes are further integrated by maximizing an energy function using efficient dynamic programming with efficient peak pattern trimming. The experimental results on popular UCF-Sports and UCF-101 datasets demonstrate the effectiveness of our approach.

UCF101	ABO	MABO	Recall	#Prop.
Brox & Malik [2]	13.28	12.82	1.40	**3**
Yu *et al.* [42]	n.a	n.a	0.0	10,000
APT [39]	40.77	39.97	35.45	2299
Li *et al.* [16]	**63.76**	40.84	39.64	18
S-CRN (RGB)	50.91	51.22	56.10	30
T-CRN (RGB)	50.52	50.89	55.10	30
TORNADO-RGB	53.10	53.30	59.59	30
S-CRN (FLOW)	46.23	46.28	44.30	30
T-CRN (FLOW)	47.54	47.44	47.29	30
TORNADO-FLOW	48.79	48.80	49.70	30
TORNADO-RGB+OF(NO TRIM)	50.44	50.71	57.92	30
TORNADO-RGB+OF	54.28	**54.93**	**61.42**	30

Table 1. Quantitative comparison on the UCF-101 dataset. Recall is computed at a precision threshold of 0.5.

UCF Sports	ABO	MABO	Recall	#Prop.
Brox & Malik [2]	29.84	30.90	17.02	**4**
Jain *et al.* [11]	63.41	62.71	78.72	1642
Oneata *et al.* [22]	56.49	55.58	68.09	3000
Gkioxari *et al.* [9]	63.07	62.09	87.23	100
APT [39]	65.73	64.21	89.36	1449
Li *et al.* [16]	**89.64**	74.19	91.49	12
S-CRN (RGB)	72.73	73.83	96.79	30
T-CRN (RGB)	69.34	70.89	94.67	30
TORNADO-RGB	73.19	74.26	96.79	30
S-CRN (FLOW)	68.32	69.29	94.67	30
T-CRN (FLOW)	67.55	67.99	92.54	30
TORNADO-OF	69.80	70.59	94.67	30
TORNADO-RGB+OF(NO TRIM)	75.23	**76.20**	**96.79**	30
TORNADO-RGB+OF	75.23	**76.20**	**96.79**	30

Table 2. Quantitative comparison on the UCF-Sports dataset. Recall is computed at a precision threshold of 0.5.

References

[1] N. Ballas, L. Yao, C. Pal, and A. C. Courville. Delving deeper into convolutional networks for learning video representations. *CoRR*, abs/1511.06432, 2015.

[2] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[3] W. Chen and J. J. Corso. Action detection by implicit intentional motion clustering. In *ICCV*, 2015.

[4] M. Cheng, Z. Zhang, W. Lin, and P. H. S. Torr. BING: binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[6] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis, 13th Scandinavian*

Conference, SCIA 2003, Halmstad, Sweden, June 29 - July 2, 2003, Proceedings, pages 363–370, 2003.

[7] H. Fu, D. Xu, and S. Lin. Object-based multiple foreground segmentation in RGBD video. *IEEE Trans. Image Processing*, 26(3):1418–1427, 2017.

[8] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[9] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[11] M. Jain, J. C. van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014.

[12] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[13] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[15] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011.

[16] N. Li, D. Xu, Z. Ying, and G. L. Zhihao Li. Search action proposals via spatial actionness estimation and temporal path inference and tracking. In *ACCV*, 2016.

[17] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek. Videolstm convolves, attends and flows for action recognition. *CoRR*, abs/1607.01794, 2016.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

[19] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff. Action recognition and localization by hierarchical space-time segments. In *ICCV*, 2013.

[20] F. Meng, H. Li, S. Zhu, B. Luo, C. Huang, B. Zeng, and M. Gabbouj. Constrained directed graph clustering and segmentation propagation for multiple foregrounds cosegmentation. *IEEE Trans. Circuits Syst. Video Techn.*, 25(11):1735–1748, 2015.

[21] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1020–1028, 2016.

[22] D. Oneata, J. Revaud, J. J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014.

[23] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *ICLR-Workshop*, 2016.

[24] X. Peng, J. Lu, Z. Yi, and Y. Rui. Automatic subspace learning via principal coefficients embedding. *IEEE Trans. Cybern.*, PP(99):1–14, 2016.

[25] X. Peng and C. Schmid. Multi-region two-stream R-CNN for action detection. In *ECCV*, 2016.

[26] M. M. Puscas, E. Sangineto, D. Culibrk, and N. Sebe. Unsupervised tube extraction using transductive learning and dense trajectories. In *ICCV*, 2015.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[28] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[29] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[30] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *ECCV*, 2016.

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

[32] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016.

[33] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[34] Z. Shou, D. Wang, and S. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[35] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013.

[36] D. Tran and J. Yuan. Optimal spatio-temporal path discovery for video event detection. In *CVPR*, 2011.

[37] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012.

[38] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[39] J. C. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek. APT: action localization proposals from dense trajectories. In *BMVC*, 2015.

[40] R. Vial, H. Zhu, Y. Tian, and S. Lu. Search video action proposal with recurrent and static yolo. *ICIP*, 2017.

[41] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015.

[42] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[43] J. T. Zhou, X. Xu, S. J. Pan, I. W. Tsang, Z. Qin, and R. S. M. Goh. Transfer hashing with privileged information. In *IJCAI*, 2016.

[44] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.