Spatio-Temporal Action Detection with Cascade Proposal and Location Anticipation

Zhenheng Yang
zhenheny@usc.edu
Jiyang Gao
jiyangga@usc.edu
Ram Nevatia
nevatia@usc.edu

Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA, USA

Abstract

In this work, we address the problem of spatio-temporal action detection in temporally untrimmed videos. It is an important and challenging task as finding accurate human actions in both temporal and spatial space is important for analyzing large-scale video data. To tackle this problem, we propose a cascade proposal and location anticipation (CPLA) model for frame-level action detection. There are several salient points of our model: (1) a cascade region proposal network (casRPN) is adopted for action proposal generation and shows better localization accuracy compared with single region proposal network (RPN); (2) action spatio-temporal consistencies are exploited via a location anticipation network (LAN) and thus frame-level action detection is not conducted independently. Frame-level detections are then linked by solving an linking score maximization problem, and temporally trimmed into spatio-temporal action tubes. We demonstrate the effectiveness of our model on the challenging UCF101 and LIRIS-HARL datasets, both achieving state-of-the-art performance.

1 Introduction

We aim to address the problem of action detection with spatio-temporal localization: given an untrimmed video, the goal is to detect and classify every action occurrence in both spatial and temporal extent. Advances in convolutional neural network (CNN) have triggered improvements in video action recognition [7, 11, 22]. Compared with action recognition, spatio-temporal action detection is more challenging due to arbitrary action volume shape and large spatio-temporal search space.

There has been previous work in spatio-temporal action detection. Recent deep learning based approaches [10, 18, 27, 32] first detect actions independently on the frame-level which are then either linked or tracked to form a final spatio-temporal action detection result. These methods use both appearance and motion features but process them separately and then fuse the detection scores. However, an action occurrence extends over a period and there should be consistency within the movement of the action regions at different temporal points. For example in the action "diving", the swinging up of arms often indicates a lunge and jump in around a second and head-down diving in another second. Thus the location of the action

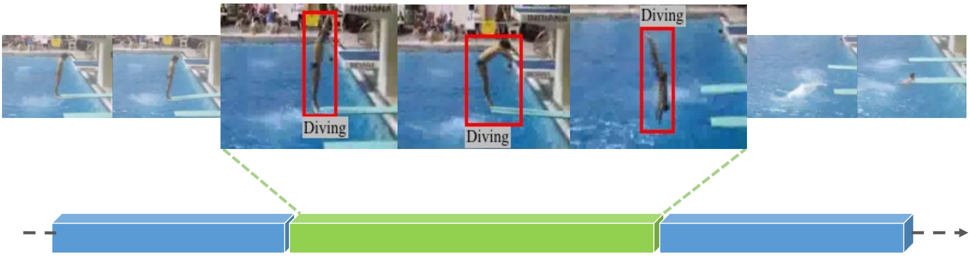

Figure 1: Spatio-temporal action detection in an untrimmed video.

in one or two seconds will be lower than the current location. With such consistencies, the relationship of action spatial localization in time can be leveraged and modeled.

We propose to use Cascade Proposal and Location Anticipation (*CPLA*) for Action Detection in videos. Specifically, our approach consists of spatial detection and temporal linking. The frame-level spatial detection model is composed of three parts: cascade RPN (*casRPN*) as action proposal generator, a network similar to fast R-CNN as the detection network and a location anticipation network (*LAN*). LAN is designed for inferring the movement trend of action occurrences between two frames, $t-K$ and t where $t>K$, K is the *anticipation gap*. LAN takes detected bounding boxes from detection network output on frame $t-K$ and then infers the corresponding boxes on frame t. The anticipated bounding boxes serve as additional proposals for the detection network on frame t. To exploit both appearance and motion cues, two-stream networks are implemented for all casRPN, detection network and LAN.

We implement the casRPN as a two stage cascade of RPNs, based on our observation that original RPN suffers from low recall at high intersection-over-union (IoU). The casRPN takes images (RGB images or optical flow images) as inputs and outputs spatial action proposals. Detection network takes proposals as input and further classifies and regresses to detection results. Spatial action detection results are temporally linked and trimmed to produce a spatio-temporal action tube, similar to [18, 27].

The proposed approach is assessed on UCF101 [25] 24 classes (UCF101-24) and LIRIS-HARL [33] datasets for proposal performance and spatio-temporal action detection performance. casRPN outperforms other proposal methods [19, 29, 35] by a large margin on both datasets, especially in the high IoU range. For action detection performance, CPLA achieves state-of-the-art performance on both datasets. For example, an mAP of 73.54% is achieved at standard spatio-temporal IoU of 0.2 on UCF101-24, an improvement of 0.68% from 72.86% , the current state-of-the-art method [18].

In summary, our contributions are twofold:

(1) We propose a location anticipation network (LAN) for action detection in videos that exploits the spatio-temporal consistency of action locations.

(2) We propose a cascade of region proposal network (casRPN) for action proposal generation which achieves better localization accuracy.

(3) We comprehensively evaluate different variants of CPLA on UCF101-24 and LIRIS-HARL datasets and CPLA achieves state-of-the-art performance.

2 Related work

Inspired by the advances in image classification and object detection on images, the deep learning architectures have been increasingly applied to action recognition, temporal action detection, spatial action detection and spatio-temporal action detection in videos.

R-CNN for Object detection. R-CNN [9] has achieved a significant success in object detection in static images. This approach first extracts proposals from images with selective search [29] algorithm and then feeds the rescaled proposals into a standard CNN network for feature extraction. A support vector machine (SVM) is then trained on these features and classifies each proposal into one of object categories or background. There are a sequence of works improving R-CNN [9]. SPP-net [12] implements a spatial pyramid pooling strategy to remove the limitation of fixed input size. Fast R-CNN [8] accelerates R-CNN by introducing a ROI pooling layer and improve the accuracy by implementing bounding box classification and regression simultaneously. Faster R-CNN [19] further improves the speed and performance by replacing proposal generation algorithm with a region proposal network (RPN).

Action spatial detection and temporal detection. There have been considerable works on spatial action detection in trimmed videos and temporal detection in untrimmed videos. On spatial action detection, Lu *et al.* [15] propose to use both motion saliency and human saliency to extract supervoxels and apply a hierarchical Markov Random Field (MRF) model for merging them into a segmentation result. Soomro *et al.* [26] further improve the performance by incorporating spatio-temporal contextual information into the displacements between supervoxels. Wang *et al.* [31] first apply a two-stream fully convolutional network to generate an action score map (called "actionness map"). Then action proposals and detections are extracted from the actionness map.

For action temporal detection, sliding window based approaches have been extensively explored [6, 28, 30]. Bargi *et al.* [2] apply an online HDP-HMM model for jointly segmenting and classifying actions, with new action classes to be discovered as they occur. Ma *et al.* [16] address the problem by applying a LSTM network to generate detection windows based on frame-wise prediction score. Singh *et al.* [24] extends two-stream networks to multi-stream LSTM networks. S-CNN [21] propose a two-stage action detection framework: first generate temporal action proposals and then score each proposal with a trained detection network.

Spatio-temporal action detection. Although there have been a lot of efforts on both spatial action detection and temporal action detection, only a handful of efforts have been devoted to the joint problem of localizing and classifying action occurrences in temporally untrimmed videos. Tian *et al.* [28] extend 2D deformable part model [4] to action detection in videos. Jain *et al.* [13] use super-voxels to find the action boundaries. More recent works leverage the power of deep learning networks. Gkioxari and Malik [10] extract proposals on RGB with selective search algorithm, and then apply R-CNN network on both RGB and optical flow data for action detection per frame. The frame-level detections are linked via Viterbi algorithm. Wainzaepfel *et al.* [32] replace selective search with a better proposal generator, i.e. EdgeBoxes [35] and conduct tracking on some selected frame-level action detections. Mettes *et al.* [17] propose to use sparse points as supervision to generate proposals. The two most recent works [18, 27] extend faster R-CNN in static images and train appearance and motion networks for frame-level action detection independently. The detections of two-stream networks are late fused and linked via Viterbi algorithm [5]. A temporal trimming is then applied to generate spatio-temporal action tubes.

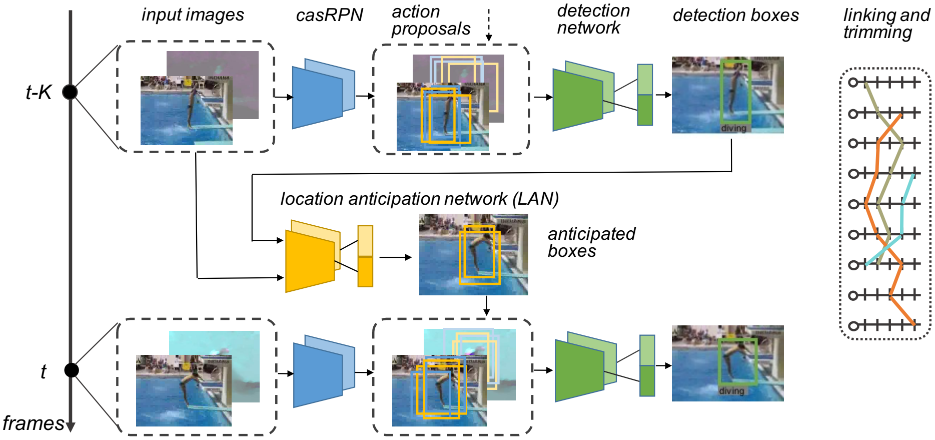

Figure 2: Overview of cascade proposal and location anticipation (CPLA) model for action detection. Inputs are RGB and optical flow images, processed by casRPN and detection network. Location anticipation network (LAN) leverages the temporal consistency by inferring action region movement. The right most part represents detection linking and trimming. The horizontal and vertical lines represent different frames and detections on each frame. Three lines linking action detections across frames represent generated spatio-temporal action tubes.

3 Methodology

As shown in Figure 2, a proposal network (casRPN), detection network and LAN are combined to generate frame-level action detections. The outputs are then linked and temporally trimmed to generate action tubes. In this section, the model architecture and training procedures are discussed in detail.

3.1 Proposal Generation Network

We adopt a two-stage cascade of RPNs [19] that we call *casRPN*. Similar to [19], each stage is built on top of the last convolutional layer of the VGG-16 network [23] followed by two sibling output layers: classification (*cls*) and regression (*reg*). To generate region proposals, the original RPN slides over the feature map output by the last convolutional layer and takes reference bounding boxes (called *"anchor boxes"*) as input, outputting the objectness score and bounding box regression coordinates. For the first stage of casRPN (*RPN-a*), we follow the anchor box generation process as in [19]. The proposals generated from RPN-a serve as the anchor boxes of the second RPN (*RPN-b*) for scoring and another round of regression. Final proposal results are *reg-b* and *cls-b* generated from RPN-b. More details of the architecture are shown in Figure 3 (a).

Training. The two stages of VGG-16 net are trained independently. A training protocol similar to [19] is followed: anchors with a high Intersection-over-Union (IoU) with the ground truth boxes (IoU > 0.7) are considered as positive samples, while those with low IoU (IoU < 0.3) as negative samples. Considering that in the action datasets (such as UCF101),

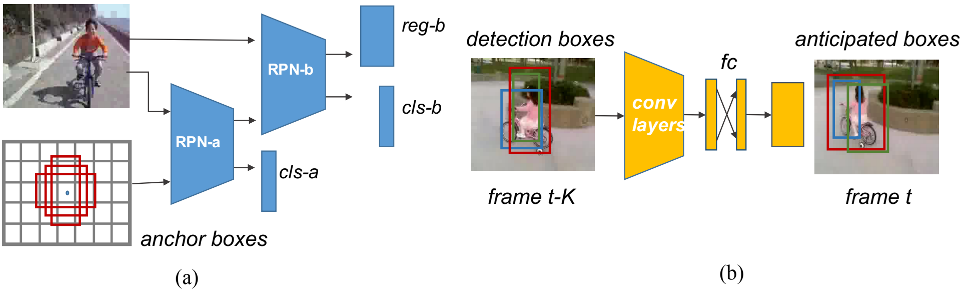

Figure 3: Network architectures: (a) casRPN and (b) LAN.

there are fewer occurrences in one frame compared to those in the object detection datasets (such as Pascal VOC 2007 [3]). To achieve network fast convergence, we ensure that in each mini-batch, the positive-negative sample ratio is between 0.8 and 1.2, and also that the mini-batch size is no larger than 128. The learning rate is set to be 0.0005. The Adam [14] optimizer is used.

3.2 Detection network

The detection network is built upon the last convolutional layer of VGG-16 network [23] followed by a fusion of two stream features, two fully connected (*fc*) layers and then two sibling layers as regression and classification. The detection network takes the images (RGB and optical flow images) and proposals as input and regresses the proposals to a new set of bounding boxes for each class. Each proposal leads to C (number of classes) bounding boxes and corresponding classification scores. The network architecture is similar to that of fast R-CNN [8] except that the two-stream features are concatenated at *conv*5 level.

Training. For detection network training, a variation of training procedure in [19] is implemented. Shaoqing *et al.* [19] introduced a four-step 'alternating training' strategy: RPN and detection network (fast R-CNN) are trained independently in the first two steps while in the 3rd and 4th step, the two networks are fine-tuned with shared convolutional weights. In our method, the casRPN and detection network are trained independently following the first two steps as we found the detection accuracy decreases when using shared convolutional weights. During training, the number of proposal bounding boxes and learning rate are set to be 2000 and 0.001. Adam optimizer [14] is employed.

3.3 Location Anticipation Network

We use the location anticipation network (LAN) to predict the movement of action occurrences within an anticipation gap K (frames). The input to LAN are RGB images and optical flow images and the action detection results on frame $t-K$. Regions of interest (ROIs) are extracted from input images and processed by each stream of LAN. The two stream features computed from RGB and optical flow images are then concatenated and the anticipated bounding boxes are generated from a regression layer applied on the fused features. The inferred action bounding boxes serve as additional proposals and fed into the detection network on frame t The LAN is built upon the last convolutional layer of fast R-CNN [8] network followed by the fusion of two stream features and fc regression layers. The architecture of LAN is illustrated in Figure 3 (b).

Two stream fusion. As the inputs to motion stream of the network are optical flow images, they already contain action movement information; we propose to leverage the coupling of appearance and motion cues to provide more prediction information by concatenating two-stream *conv5* features before fc layers in LAN.

Training. For LAN, the input contains the detection results on frame $t-K$ ($t > K$) and the target is the ground truth bounding boxes of frame t. Similar to the training protocol for casRPN 3.1, bounding boxes in frame t having high IoU (IoU>0.7) with frame t ground truth boxes are considered positive samples while low IoU (IoU < 0.3) are taken to be negative samples. The loss function for an image is defined as:

$$L(t_i) = \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{1}$$

where, i is the index of a detection bounding box, t_i is a vector representing the parameterized bounding box coordinates, t_i^* is the associated ground truth bounding box, N_{reg} is the number of detection bounding boxes, the ground truth label term p_i^* is 1 if the detection box is positive and 0 if it is negative. The ground truth label $L_{reg}(t_i, t_i^*)$ is a smooth L_1 loss function as in [20]. The mini-batch size and learning rate are set to be 300 and 0.001 separately and Adam optimizer [14] is deployed for minimizing the loss function above.

3.4 Detection linking and temporal trimming.

We employ detection linking using the Viterbi algorithm [5] and apply maximum subarray algorithm for temporal trimming as in [18].

Detection linking. The linking score function $S_c(d_t, d_{t+1})$ of linking action detections of class c in two consecutive frames is defined as below

$$S_c(d_t, d_{t+1}) = (1-\beta) * (s_c(d_t) + s_c(d_{t+1})) + \beta * \Psi(d_t, d_{t+1}) \tag{2}$$

in which, $s_c(d_i)$ is the class score of detection box at frame i, $\Psi(d_t, d_{t+1})$ is the IoU between two detection boxes, β is a scalar weighting the relative importance of detection score and overlaps. The linking score is high for those links in which detection boxes score high for the action class c and also overlap highly in consecutive frames. We find paths with maximum linking scores via Viterbi algorithm [5]. β is empirically set to be 0.7.

Temporal trimming. In realistic videos, there is no guarantee that human actions occupy the whole span thus temporal trimming is necessary for spatio-temporal action localization. Inspired by [1], we employ an optimal subarray algorithm similar to [18]. Given a video track Γ, we aim to find a subset starting from frame s to frame e within Γ. The optimal subset $\Gamma_{(s,e)}$ maximizes the objective:

$$\frac{1}{(e-s)} \sum_s^e S_c(d_t, d_{t+1}) - \frac{(e-s) - \overline{L_c}}{\overline{L_c}} \tag{3}$$

In this objective function, s and e are the indexes of starting and ending frame of the track subset. $S_c(d_t, d_{t+1})$ is the linking score as in 2. $\overline{L_c}$ is the average length (in frames) of action category c in training data. The objective function aims to maximize the average linking scores between two frames in a track and to minimize the drift of generated track length from the average track length.

4 Evaluation

We first introduce the datasets and evaluation metrics used in our experiments and then present a comprehensive evaluation of our methods.

4.1 Datasets and metrics.

Datasets. Two widely used datasets are selected for evaluating the performance of our action proposals and spatio-temporal action detection: (1) UCF101 [25] 24 classes (UCF101-24) and (2) LIRIS-HARL [33].

(1) UCF101-24 is a subset of larger UCF101 action classification dataset. This subset contains 24 action classes and 3207 temporally-untrimmed videos, for which spatio-temporal ground truths are provided. As in [18, 27], we evaluate on the standard training and test split of this dataset. (2) LIRIS-HARL dataset contains temporally-untrimmed 167 videos of 10 action classes. Similar to the UCF101-24 dataset, spatio-temporal annotations are provided. This dataset is more challenging, as there are more action co-occurrences in one frame, more interacting actions with humans/objects, and cases where relevant human actions take place among other irrelevant human motion.

Evaluation metrics. The evaluation metrics in original papers [25, 33] are followed for separate dataset. Specifically, for assessing the quality of proposals, *recall-vs-IoU* curve is employed. When measuring the action detection performance on UCF101-24, *mean Average Precision* (*mAP*) at different spatio-temporal IoU threshold of $\{0.05,0.1,0.2,0.3,0.4,0.5\}$ is reported, using the evaluation code of [27]. The official evaluation toolbox of LIRIS-HARL is used to measure the spatial precision and recall, along with temporal precision and recall, and finally integrating into a final score. Different metrics are reported: $Recall_{10}$, $Precision_{10}$, F1-$Score_{10}$, etc. More details of evaluation metrics are available in the original paper. The mAP metric performance as in UCF101-24 is also presented for better comparison.

4.2 Experiments

Several experiments are conducted on UCF101-24 and LIRIS-HARL dataset for an comprehensive evaluation of CPLA approach: (1) casRPN is compared with other proposal methods for proposal quality assessment; (2) Two different anticipation strategies and non-anticipation model are compared. (3) Different anticipation gaps K are explored and discussed. (4) Two fusion methods are explored and compared. (5) The CPLA model is compared with state-of-the-art methods on UCF101-24 and LIRIS-HARL datasets in terms of spatio-temporal action detection performance.

Evaluation of casRPN performance. The quality of casRPN proposals are compared with several other proposal methods on UCF101-24 split1. These methods include Selective Search (SS) [29], EdgeBoxes(EB) [35], RPN trained on ImageNet, RPN trained on UCF101-24 split1 (both appearance and motion models, RPN-a, RPN-m) and casRPN trained on UCF101-24 split1 (casRPN-a, casRPN-m). For Selective Search and EdgeBoxes, the default settings are implemented (2000 and 1000 proposals are extracted separately from one image). While for RPN based methods, top 300 proposals are picked. The recall-vs-IoU curves are plotted for evaluating proposal quality. Figure 4 shows that even with a relatively smaller number of proposals, RPN based proposals consistently exhibit much better recall performance compared to the two non deep-learning methods. casRPN outperforms other

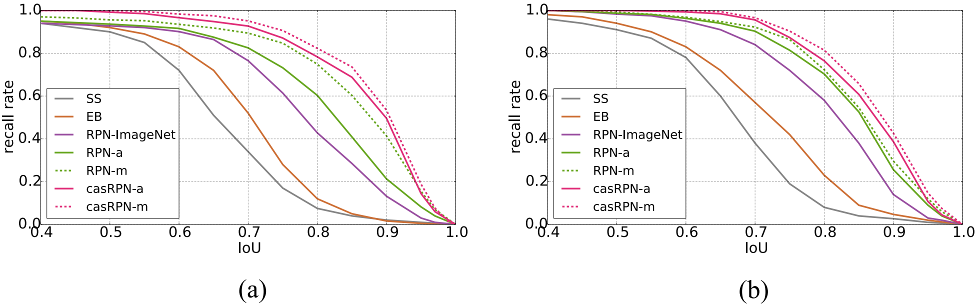

Figure 4: Performance comparison of action proposal methods on (a) UCF101-24 split1 and (b) LIRIS-HARL.

RPN methods by a large margin, especially at high IoU region, validating the effectiveness of two stage of regression in accurate localization.

Anticipation strategy study. We compare different anticipation strategies: (1) Original CPLA model. (2) The detection bounding boxes from frame $t-K$ are directly fed as proposals to detection network of frame t (Non-motion CPLA). In this case, zero motion is assumed during the prediction gap K. (3) Detections on different frames are conducted independently and the proposals of each frame only come from the casRPN (Non-anticipation model). For all three different anticipation models, casRPN is employed as proposal generator and the performances are compared with mAP metric on the testing part of UCF101-24 split 1. As shown in Table 1, the models that exploit the spatio-temporal consistency of actions locations benefit from the additional information compared with independent frame-level detection. At the standard threshold of $\delta = 0.2$, CPLA achieves mAP of 73.54%. Comparing the two different anticipation strategies, we can see that a trained anticipation model (CPLA) outperforms the naive non-motion model (Non-motion CPLA) by 3.11% at $\delta = 0.2$ when $K = 8$. These comparisons indicate that anticipation model leveraging the consistency of action locations in temporal space helps boost the performance of spatio-temporal action detection. We also explored use of multiple frames within the anticipation gap and feeding their detection results as inputs to LAN. Uniformly sampling 2 frames from the anticipation gap shows 2.4% mAP performance drop on UCF101-24 at $\delta = 0.2$. This can be explained that doubling the input size, the number of parameters of fc layer also double and thus LAN may be overfitting.

Exploring different anticipation gaps. Different choices of the anticipation gaps K are also explored: K is set to be $\{2,8,16\}$ respectively to see how it affects the mAP performance. As shown in Table 1, $K = 8$ shows the best result and outperforms the other anticipation gaps by at least 2.50% at $\delta = 0.2$. The performance decreases with too short ($K = 2$) or too long ($K = 16$) gaps. For too short anticipation gap, the network is predicting the movement between a very short duration of 0.06s. No obvious motion happens in this short time and thus it shows similar performance to zero motion model. With too long an anticipation gap, the performance drops. It can be explained by the observation that temporal cues are too noisy when looking at a relatively long time before the action occurs.

Modeling of spatio-temporal cues. The optical flow images contain movement trend and can be leveraged for prediction along with the appearance information. We compare two

Table 1: Action detection performances under different anticipation strategies

Spatio-temporal overlap threshold (δ)		0.05	0.1	0.2	0.3
Non-anticipation model		77.32	76.71	66.35	56.73
Non-motion CPLA	$K = 2$	77.45	76.31	67.23	56.84
	$K = 8$	78.86	77.01	70.43	59.24
	$K = 16$	76.53	75.02	66.65	55.63
CPLA	$K = 2$	78.25	76.28	70.82	59.38
	$K = 8$	**79.03**	**77.34**	**73.54**	**60.76**
	$K = 16$	77.84	75.83	71.04	58.92

different methods of modeling the fusion of two-stream cues: (1)Boost appearance detection results with motion detection scores as in [27], i.e. model the two-stream coupling on the bounding box level (CPLA-bbox); (2) Couple the appearance and motion information at the feature level, i.e. concatenate two stream *conv5* features (CPLA-conv5). As shown in Table 2, with explicitly modeled two-stream feature coupling, CPLA-conv5 outperforms CPLA-bbox consistently, which only uses the fusion to assist appearance detection.

Table 2: Two variants of CPLA using different modeling of two-stream cues

δ	0.05	0.1	0.2	0.3
CPLA-bbox	77.33	74.72	70.39	58.87
CPLA-conv5	**79.03**	**77.34**	**73.54**	**60.76**

Analysis on detection linking hyperparameter. The hyperparatmer β in Equation 2 affects the trade-off between detection score and spatial overlap (IoU) in the linking process. β is set to be 0.7 empirically. Higher β gives more weight on the relative importance of IoU and leads to more fragmented linking. Lower β leads to ID switches in the linking. Both result in lower mAP.

Comparison with state-of-the-art. The comparisons with state-of-the-art methods on UCF101-24 and LIRIS-HARL datasets are presented in Table 3 and Table 4 separately. On UCF101-24, CPLA outperforms [18] by 0.68% at spatio-temporal IoU = 0.2. On LIRIS-HARL, CPLA outperforms the current state-of-the-art methods by a large margin under both evaluation protocols. Some qualitative results are shown in Figure 5 on UCF101-24 video.

Table 3: Quantitative action detection results on UCF101-24 dataset comparing with state-of-the-art methods.

δ	0.05	0.1	0.2	0.3	0.4	0.5
FAP[34]	42.80	-	-	-	-	-
STMH[32]	54.28	51.68	46.77	37.82	-	-
Saha *et al.* [27]	**79.12**	76.57	66.75	55.46	46.35	35.86
MR-TS R-CNN [18][1]	78.76	77.31	72.86	**65.70**	-	-
CPLA	79.03	**77.34**	**73.54**	60.76	**49.23**	**37.80**

[1] Updated results of [18] from https://hal.inria.fr/hal-01349107/file/eccv16-pxj-v3.pdf

Table 4: Quantitative action detection results on LIRIS-HARL dataset under different metrics

Methods	Recall$_{10}$	Precision$_{10}$	F1-Score$_{10}$	I_{sr}	I_{sp}	I_{tr}	I_{tp}	IQ	mAP@$\delta = 0.2$
Saha *et al.* [□]	0.57	0.60	0.58	0.54	0.34	0.48	**0.47**	0.46	49.10
CPLA	**0.62**	**0.67**	**0.70**	**0.62**	**0.42**	**0.51**	0.44	**0.53**	**54.34**

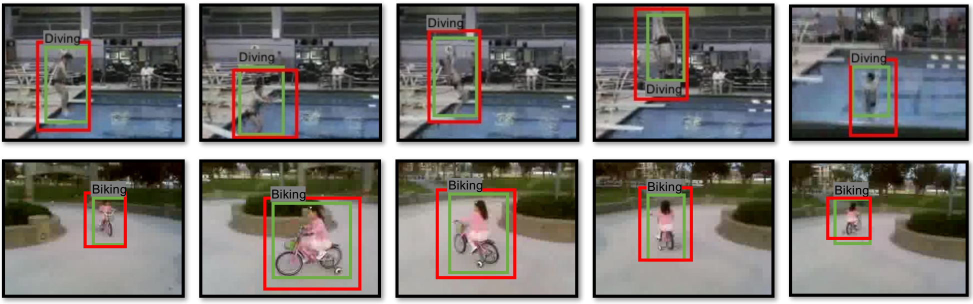

Figure 5: Qualitative results on UCF101-24 video. Red bounding boxes are annotations and green bounding boxes are action detection results

5 Conclusion

This paper introduced a cascade proposal and location anticipation (CPLA) model for spatio-temporal action detection. CPLA consists of a frame-level action detection model and a temporal linking/trimming algorithm. The action detection model takes RGB and optical flow images as input, extracts action proposals via casRPN and conducts action detection on each frame by exploiting the action region movement continuity. CPLA achieves state-of-the-art performance on both UCF101 and more challenging LIRIS-HARL datasets.

References

[1] Senjian An, Patrick Peursum, Wanquan Liu, and Svetha Venkatesh. Efficient algorithms for subwindow search in object detection and localization. In *CVPR*, pages 264–271, 2009.

[2] Ava Bargi, Richard Yi Da Xu, and Massimo Piccardi. An online hdp-hmm for joint action segmentation and classification in motion capture data. In *CVPR Workshop*, pages 1–7. IEEE, 2012.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[4] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8. IEEE, 2008.

[5] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[6] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2782–2795, 2013.

[7] Chuang Gan, Chen Sun, Lixin Duan, and Boqing Gong. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *ECCV*, pages 849–866, 2016.

[8] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[10] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015.

[11] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *ICCV*, pages 1080–1088, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361, 2014.

[13] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, 2014.

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[15] Jiasen Lu, Jason J Corso, et al. Human action segmentation with hierarchical supervoxel consistency. In *CVPR*, pages 3762–3771, 2015.

[16] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, pages 1942–1950, 2016.

[17] Pascal Mettes, Jan C van Gemert, and Cees GM Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, pages 437–453, 2016.

[18] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, pages 744–759, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[20] Mark Schmidt, Glenn Fung, and Rmer Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *ECCV*, pages 286–297, 2007.

[21] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[22] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, pages 1961–1970, 2016.

[25] 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.

[26] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Action localization in videos through context walk. In *CVPR*, pages 3280–3288, 2015.

[27] Michael Sapienza Philip H. S. Torr Fabio Cuzzlion Suman Saha, Gurkirt Singh. Deep learning for detecting multiple space-time action tubes in videos. *BMVC*, 2016.

[28] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, pages 2642–2649, 2013.

[29] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[30] Limin Wang, Yu Qiao, and Xiaoou Tang. Video action detection with relational dynamic-poselets. In *ECCV*, pages 565–580, 2014.

[31] Limin Wang, Yu Qiao, Xiaoou Tang, and Luc Van Gool. Actionness estimation using hybrid fully convolutional networks. In *CVPR*, pages 2708–2717, 2016.

[32] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *CVPR*, pages 3164–3172, 2015.

[33] Christian Wolf, Eric Lombardi, Julien Mille, Oya Celiktutan, Mingyuan Jiu, Emre Dogan, Gonen Eren, Moez Baccouche, Emmanuel Dellandréa, Charles-Edmond Bichot, et al. Evaluation of video activity localizations integrating quality and quantity measurements. *Computer Vision and Image Understanding*, 127:14–30, 2014.

[34] Gang Yu and Junsong Yuan. Fast action proposals for human action detection and search. In *CVPR*, pages 1302–1311, 2015.

[35] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014.