MULTIPLE PATH SEARCH FOR ACTION TUBE DETECTION IN VIDEOS

Erick Hendra Putra Alwando, Yie-Tarng Chen, and Wen-Hsien Fang

Department of Electronic and Computer Engineering
National Taiwan University of Science and Technology
Email:{M10402805, ytchen, whf}@mail.ntust.edu.tw

ABSTRACT

This paper presents an efficient convolutional neural network (CNN)-based multiple path search (MPS) algorithm to detect multiple spatial-temporal action tubes in videos. With the pass information and the accumulated scores generated by forward message passing, the new algorithm reuses these information to simultaneously find multiple paths in backward path tracing without repeating the search process. Moreover, to rectify the potentially inaccurate bounding boxes, we also propose a video localization refinement scheme to further boost the detection accuracy. Simulations show that the proposed algorithm provides competing performance compared with the main state-of-the-art works on the widespread UCF-101 dataset with yet lower complexity.

Index Terms— Action localization, convolutional neural networks (CNN), multiple path search, localization refinement, object detection.

1. INTRODUCTION

With the advancement of object detection techniques [1, 2, 3], action localization in videos, which is at the core of a variety applications such as self-driving car, humanoid robot, etc [4, 5, 6, 7], has become an emerging research topic. However, due to the complicated nature of occlusion, low resolution of video, non-steady view point, etc, automatic action localization, especially for multiple objects, is a challenging task.

A myriad of algorithms have been addressed for efficaciously and efficiently localize the action in the videos. For example, the temporal sliding window approach was utilized in [8, 9, 10, 11], where the handcrafted features were used for classification. However, these approaches do not employ the potent convolutional neural network (CNN) features and thus their performance is in general inferior to the state-of-the-art methods. The spatio-temporal information were obtained based on the CNN features in [6] and the bounding boxes are then linked together by a learning process using the SVM classifier. However, their complexity are not light due to the expensive learning process. Yu *et al.* [7] formulated the action proposal generation as a maximum set coverage problem and conducted the greedy approach to distinguish all possible paths in a video. Recently, Saha *et al.* [4] considered a dynamic programming (DP) approach to detect multiple action tubes in videos, which extended [5] from single to multiple action scenarios. Nevertheless, [4] did not fully utilize the information generated in each iteration, so it has to re-start the search to find one action path after another and thus is computationally demanding, especially for the videos that have lots of instances and frames.

This paper presents an efficient two-stream faster R-CNN based action localization algorithm, termed multiple path search (MPS), to detect multiple spatial-temporal action tubes in videos. Similar as the standard DP, MPS first finds some possible associations of spatially-nearby and temporal-continuous detection boxes in the forward message passing. Additionally, a newly-developed backward path tracing is invoked thereafter to *simultaneously* find multiple paths by fully reusing the information generated in the forward pass without repeating the search process. Thereby, the complexity incurred can be reduced. Moreover, since the inaccurate bounding boxes produced by faster R-CNN may impact the detection accuracy, we also propose a video localization refinement (VLR) scheme to iteratively rectify the bounding boxes. Together with VLR, the performance of the proposed algorithm can be further boosted. Simulation results show that the proposed algorithm provides competing performance compared with the main state-of-the-art works on the widespread UCF-101 dataset with yet lower complexity.

The contributions of this paper include: 1) a novel low-cost DP-like algorithm is developed to efficiently find *multiple* paths with only a single run; 2) an iterative VLR scheme, which can rectify the potentially inaccurate bounding boxes generated by the two-stream faster R-CNN into their correct position and scales, is addressed to further enhance the detection accuracy.

2. PROPOSED ALGORITHM

In this section, we begin with the introduction of the overall structures of the proposed action localization method. Thereafter, we focus on two new building blocks - video location refinement scheme and multiple path search algorithm.

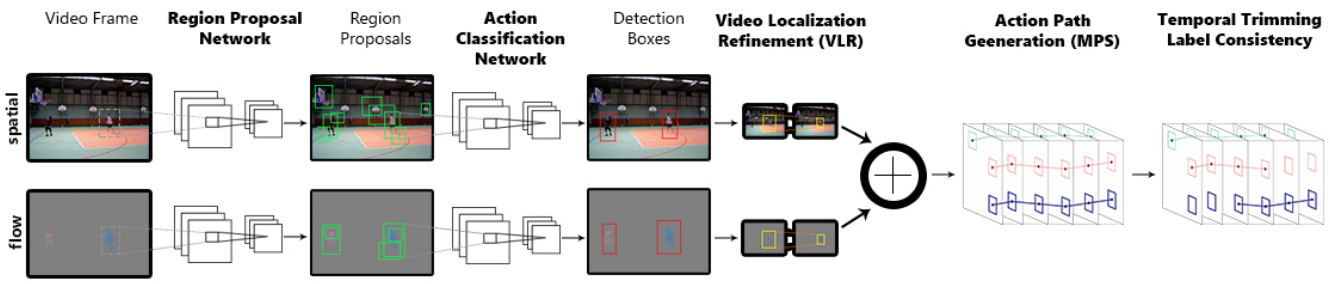

Fig. 1. Overview of the proposed action localization framework.

2.1. Overall Methodology

Our CNN-based action localization method consists of the following five steps, the first two and the last one of which are the same as the existing CNN-based action localization systems [4]. The third step is a new iterative bounding box refinement scheme while the fourth step is the proposed multiple search algorithm. For easy reference, the overall procedures of the proposed method are as illustrated in Fig. 1.

Region Proposal Network: Firstly, region proposals are generated [1] to facilitate action localization in later stages.

Action Classification Network: Secondly, each of the generated region proposals is extracted using a two-stream faster R-CNN [1] model to get the deep features, and a softmax score is given for each specific action class.

Video Localization Refinement: Thirdly, VLR is invoked for both of the spatial and flow domains of the two-stream faster R-CNN to refine the possibly inaccurate bounding boxes. Finally, a fusion strategy is employed to combine these two detection scores.

Multiple Path Generation: Fourthly, the proposed MPS algorithm is employed to find the multiple paths.

Temporal Trimming Label Consistency: Finally, temporal label trimming consistency is invoked to ensure the background and action label in the temporal domain by analyzing the action path scores. If the sequence of scores are consistent and smooth, then it is categorized as an action.

2.2. Video Localization Refinement

To rectify the detection boxes generated by the detection network into their correct positions and scales, to follow we consider a VLR method, which is an extension of [12] from images to videos. Let $\mathbf{x}_{i,j}$ denote the detection box for the j^{th} object from either the spatial or flow domains generated from faster R-CNN in frame i with a detection score $s_c(\mathbf{x}_{i,j})$ for a particular class c. In each iteration, VLR first finds a new bounding box in the search region, in which the bounding boxes have detection scores greater than that determined in the previous iteration. Thereafter, inspired by the mean shift algorithm [13], we shift the center of the search region to the new bounding box and repeat the iteration. The overall VLR consists of the following two steps in each iteration:

Determination of search region:

We begin with deciding the search region $\mathcal{X}_{i,j}$ for each input detection box $\mathbf{x}_{i,j}$ by considering the region proposals from the the current frame and its neighboring frames, *i.e.* frames $(i-1)$ and $(i+1)$. Let $\mathcal{Q}$ denote the entire set of region proposals and $\mathcal{N}(\mathbf{x}_{i,j}) = \{\mathbf{z}|\mathsf{IoU}(\mathbf{z},\mathbf{x}_{i,j}) \geq \gamma, \mathbf{z} \in \mathcal{Q}\}$ be a subset of $\mathcal{Q}$ that overlaps with $\mathbf{x}_{i,j}$ with $\gamma \in [0,1]$ being the degree of overlapping, where $\mathsf{IoU}(\mathbf{x}_1,\mathbf{x}_2) = \frac{\mathsf{area}(\mathbf{x}_1 \cap \mathbf{x}_2)}{\mathsf{area}(\mathbf{x}_1 \cup \mathbf{x}_2)}$ [14]. The search region $\mathcal{X}_{i,j}$ for each detection box $\mathbf{x}_{i,j}$ is bounded by the smallest bounding box $\mathbf{z}_\cap$ and the largest bounding box $\mathbf{z}_\cup$ in $\mathcal{N}(\mathbf{x}_{i,j})$, where $\mathbf{z}_\cap = \cap_{\mathbf{z}\in\mathcal{N}(\mathbf{x}_{i,j})}\mathbf{z}$ and $\mathbf{z}_\cup = \cup_{\mathbf{z}\in\mathcal{N}(\mathbf{x}_{i,j})}\mathbf{z}$.

Localization refinement:

Next, we attempt to find a new bounding box $\mathbf{r}_{i,j}$ in $\mathcal{X}_{i,j}$, $\mathcal{X}^*_{i,j}$, which has a detection score greater than $s_c(\mathbf{x}_{i,j})$ and the detection scores of the bounding boxes in $\mathcal{X}^*_{i,j}$, where $\mathcal{X}^*_{i,j} = \{\mathbf{y}|\mathbf{y} \in \mathcal{X}_{i,j}, s_c(\mathbf{y}) > s_c(\mathbf{x}_{i,j})\}$ denotes denotes a set of bounding boxes in $\mathcal{X}_{i,j}$ with scores greater than $s_c(\mathbf{x}_{i,j})$. To facilitate the search, we define a new probability associated with the subset $\mathcal{X}^*_{i,j}$ as $Prob(\mathcal{X}^*_{i,j}) = |\mathcal{X}^*_{i,j}|/|\mathcal{X}_{i,j}|$, which estimates how likely the desired bounding box $\mathbf{r}_{i,j}$ exists inside $\mathcal{X}^*_{i,j}$ and $|\cdot|$ denotes the cardinality of a set. Thereby, we can determine a new bounding box $\mathbf{r}_{i,j}$ based on $Prob(\mathcal{X}^*_{i,j})$. Thereafter, we shift the center of search region from $\mathbf{x}_{i,j}$ to $\mathbf{r}_{i,j}$ and repeat the above steps to refine the bounding box until no better bounding box is available, *i.e.* $Prob(\mathcal{X}^*_{i,j}) = 0$. Note that the detection scores form an monotonically increasing sequence based on the developed iterative scheme, so the convergence of the proposed VLR is ensured as a bounded, monotonically increasing sequence will converge [15].

2.3. Multiple Path Search Algorithm

Denote the refined detection boxes in the spatial and flow domains of faster R-CNN in frame i generated by VLR as $\mathbf{r}^s_{i,j}$ and $\mathbf{r}^f_{i,j}$, respectively. To track a path, the input is $\mathbf{r}_{i,j} = \mathbf{r}^s_{i,j}$ with a detection score $s^*_c(\mathbf{r}_{i,j})$ which is a fusion of the scores

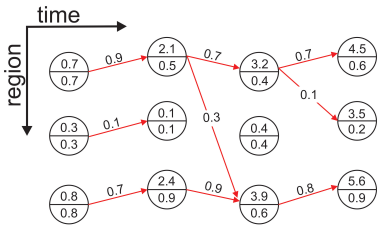

Fig. 2. A forward message passing example: A node denotes a region $\mathbf{r}_{i,j}$ with upper value $S_{i,j}$, lower value $s_c(\mathbf{r}_{i,j})$, and the value above the arrow denotes IoU. An arrow pointing from node A to B indicates that A in frame $(i-1)$ is the neighbor of B in the frame i, and A further passes message to B if this arrow is solid.

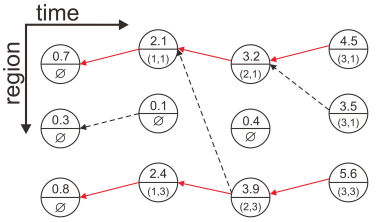

Fig. 3. A backward path tracing example: A path is traced back with solid arrows. The dotted arrow indicates a failed connection since the pointed node is used by the other path.

of the spatial and flow bounding boxes given by

$$s_c^*(\mathbf{r}_{i,j}) = s_c(\mathbf{r}_{i,j}^s) + s_c(\mathbf{r}_{i,max}^f) \times \mathsf{IoU}(\mathbf{r}_{i,j}^s, \mathbf{r}_{i,max}^f), \quad (1)$$

where $\mathsf{IoU}(\mathbf{r}_{i,j}^s, \mathbf{r}_{i,max}^f)$ denotes the intersection-over-union between $\mathbf{r}_{i,j}^s$ and $\mathbf{r}_{i,max}^f$, in which $\mathbf{r}_{i,max}^f := \{\mathbf{r}_{i,n}^f | \max\{\mathsf{IoU}(\mathbf{r}_{i,j}^s, \mathbf{r}_{i,n}^f)\} | \forall n\}$.

Denote a video path as T, which is a collection of refined detection boxes $\{\mathbf{r}_{i,j}\}$ in each frame. Our objective is to maximize the accumulative score given by

$$\sum_{T_1,\cdots,T_N} \sum_{i=1}^{M-1} A_c(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}), \quad (2)$$

where

$$A_c(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}) = s_c^*(\mathbf{r}_{i+1,l}) + \alpha \times \mathsf{IoU}(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}), \quad (3)$$

in which the subscripts j and l are dictated by the paths $\{T_n\}_{n=1}^N$, N is the number of paths, M is the total number of frames and α is the weighting parameter.

A DP approach was addressed in [4] to solve the multiple path problem. However, since the standard DP is only devised to find the optimal single path, the extension in [4] has to iteratively re-start DP to find one path after another and thus is very inefficient. To efficaciously deal with the multiple path search problem, we propose a new DP-like approach as described in Algorithm 1. In the forward message passing as shown in Fig. 2, each node stores nodes in the previous stages with the K largest accumulated scores instead of only the one with the largest score, where K is a parameter leveraging the complexity and performance. The accumulated scores and the connections of all possible paths are kept in matrices $\mathbf{S}$ and $\mathbf{P}$, respectively. In the backward path tracing as shown in Fig. 3, with $\mathbf{S}$ and $\mathbf{P}$ as inputs, we sort $\mathbf{S}$ in (2) in a descending order. We then backtrack a node with the largest score first. If it has already been assigned to any path, we begin to consider the node with the second largest score that was stored in matrix $\mathbf{P}$. Finally, we flag each path as an action if its accumulated score exceeds the threshold c.

Algorithm 1 The Multiple Path Search Algorithm

Input: $\{\mathbf{r}_{i,j}\}$ **(a set of bounding boxes)**
Output: $\mathcal{T} = \{T\}$ (a set of possible paths)

function FORWARD-MESSAGE-PASSING($\{\mathbf{r}_{i,j}\}$)
  set $S_{1,j} = s_c^*(\mathbf{r}_{1,j}), j = 1, \ldots, N$
  set $P_{i,j} = \varnothing, i = 1, \ldots, M, j = 1, \ldots, N$
  for $i = 2$ to M **do**
    for $j = 1$ to N **do**
      sort$\{\mathsf{A}(\mathbf{r}_{i-1,k^j}, \mathbf{r}_{i,j}) | k = 1, ..., K\}$
      and return $\mathsf{A}(\mathbf{r}_{i-1,k_1^j}, \mathbf{r}_{i,j}) \geq \ldots$
      $\geq \mathsf{A}(\mathbf{r}_{i-1,k_K^j}, \mathbf{r}_{i,j})$
      set $\mathbf{k}^j = \{k_1^j, \ldots, k_K^j\}$
      if $\mathbf{k}^j \neq \varnothing$ **then**
        set $S_{i,j} = S_{i-1,k_1^j} + A(r_{i-1,k_1^j}, \mathbf{r}_{i,j})$
        set $P_{i,j} = \mathbf{k}^j$
      else
        set $S_{i,j} = s_c^*(\mathbf{r}_{i,j})$
      end if
    end for
  end for
  return S,**P** (both are $M \times N$ matrices)
end function
function BACKWARD-PATH-TRACING($\mathbf{S}, \mathbf{P}, \{\mathbf{r}_{i,j}\}$)
  sort $\mathbf{S}$ and return $S_{i_1,j_1} \geq \ldots \geq S_{i_{MN},j_{MN}}$
  for $m = 1$ to MN **do**
    initialize $T = \varnothing$ and $(i, j) = (i_m, j_m)$
    while $P_{i,j} \neq \varnothing$ **do**
      $T = T \cup \mathbf{r}_{i,j}$ and $p = 1$
      while $(i-1, k_p^j)$ is assigned **do**
        $p = p + 1$
      end while
      set $(i', j') = (i - 1, k_p^j)$
      assign $(i-1, k_p^j)$ and $(i, j) = (i', j')$
    end while
    $\mathcal{T} = \mathcal{T} \cup T$
  end for
  return $\mathcal{T}$
end function

Table 1. Comparison of the action localization results on the UCF-101 dataset with different IoUs. The best results are marked in bold.

Method	0.05	0.1	0.2	0.3	0.4
FAP [7]	42.80				
STMH [6]	54.28	51.68	46.77	37.82	
Saha *et al* [4]	79.12	76.57	66.75	55.46	46.35
MPS	79.77	76.52	66.42	56.44	46.56
MPS + VLR	**83.70**	**80.08**	**68.98**	**57.08**	**46.90**

Table 2. Comparison of the search speed (frame per second).

Method	Time to finish	Frame per second
Saha *et al* [4]	37.00 min	72 fps
MPS	26.35 min	101 fps

Fig. 4. Some action detection results on UCF-101. Detection results by MPS and [4] are in green and red, respectively.

3. EXPERIMENTAL RESULTS

In this section, we compare the proposed algorithm with the main state-of-the-art methods in terms of the detection accuracy and computation time based on the popular UCF-101 action detection dataset [16], which contain a variety of action activities from different points of view.

3.1. Evaluation Protocol and Experimental Setup

The simulations mainly follow the protocols and evaluation metrics provided by the UCF-101 dataset [16] and the CNN-based action localization method [4]. The parameter α in (3) is determined using grid search with cross-validation to attain the optimal performance and $K = 5$ is employed. For the proposed VLR, we use $\gamma = 0.7$ to control the degree of overlapping (IoU) in the determination of the search region.

3.2. Performance Evaluation

In this subsection, we compare the proposed MPS method with three recently reported baselines, FAP [7], STMH [6], and [4], in terms of the detection accuracy (mAP). The comparison results are shown in Table 1, from which we can notice that Saha *et al.* [4] outperforms FAP [7] and STMH [6] for different spatio-temporal IoUs as defined in Sec. 2.2, as [4] used faster R-CNN to get detection boxes and the potent DP algorithm to find the multiple paths. The proposed MPS yields similar performance compared with [4]. This is because the new algorithm finds all multiple paths at the same time instead of separately, although it reduces the search area in each iteration. Also, we can observe that together VLR to provide more precise detection boxes, the performance of MPS can be further boosted. As an illustration, some detection results are also furnished in Fig. 4, from which we can note that the detection boxes based on the proposed MPS are more accurate compared with those based on [4].

3.3. Computation Time Analysis

Next, we compare the search speed of the proposed MPS method and the DP algorithm in [4] based on the whole UCF-101 testing data, which consists of 914 videos with 160055 total frames, as shown in Table 2. Here we measure the search speed for multiple-path generation with only 5 bounding boxes in each frame. The speed of VLR is about 3 frame per second. From Table 2 we can see that MPS is about 30 % faster than [4]. This is because MPS only conducts the search process once instead of several times as that in [4]. The complexity savings will become more significant when there are more bounding boxes and frames.

4. CONCLUSIONS

This paper has developed an efficient DP-based MPS algorithm for multiple action tube detection using CNN cues. MPS fully reuses the accumulated scores and the path information decided in the forward message passing to simultaneously find all multiple paths in backward path tracing. Moreover, a new VLR scheme is addressed to rectify the inaccurate bounding boxes, which in turn enhances the detection accuracy of MPS. Conducted simulations on the UCF-101 dataset validate the proposed method.

5. ACKNOWLEDGEMENT

This work was supported by National Ministry of Science and Technology, R.O.C. under contracts MOST 105-2221-E-011-046 and 105-2221-E-011-117.

6. REFERENCES

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Neural Information Processing Systems (NIPS)*, pp. 91–99, 2015.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[3] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016, pp. 779–788.

[4] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H. S. Torr, and Fabio Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *Proceedings of the British Machine Vision Conference*, 2016.

[5] Georgia Gkioxari and Jitendra Malik, "Finding action tubes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 759–768.

[6] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid, "Learning to track for spatio-temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3164–3172.

[7] Gang Yu and Junsong Yuan, "Fast action proposals for human action detection and search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1302–1311.

[8] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman, "Human focused action localization in video," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 219–233.

[9] Dan Oneata, Jakob Verbeek, and Cordelia Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1817–1824.

[10] Yicong Tian, Rahul Sukthankar, and Mubarak Shah, "Spatiotemporal deformable part models for action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2642–2649.

[11] Zhixin Shu, Kiwon Yun, and Dimitris Samaras, "Action detection with improved dense trajectories and sliding window," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 541–551.

[12] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang, "Improved object detection with iterative localization refinement in convolutional neural networks," in *Proceedings of the IEEE International Conference on Image Processing*, 2016, pp. 3643–3647.

[13] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[14] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al., "The pascal visual object classes challenge 2007 (voc2007) results," 2007.

[15] Edwin KP Chong and Stanislaw H Zak, *An Introduction to Optimization*, John Wiley & Sons, 2013.

[16] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CRCV-TR-12-01*, 2012.