# CNN-Based Multiple Path Search for Action Tube Detection in Videos

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

***Abstract*—This paper presents an effective two-stream convolutional neural network (CNN)-based approach to detect multiple spatial-temporal action tubes in videos. A novel video localization refinement (VLR) scheme is first addressed to iteratively rectify the potentially inaccurate bounding boxes by exploiting the temporal consistency between adjacent frames. Then, to provide more faithful detection scores, a new fusion strategy is considered, which combines not only the appearance and the flow information of the two-stream networks but also the motion saliency, the latter of which is included to address the small camera motion. Additionally, an efficient multiple path search (MPS) algorithm is developed to simultaneously identify multiple paths in a single run. In the forward message passing of MPS, each node stores information of a prescribed number of connections based on the accumulated scores determined in the previous stages. A backward path tracing is invoked afterward to find all multiple paths at the same time by fully reusing the information generated in the forward pass without repeating the search process. Thus, the complexity incurred can be reduced. The simulation results show that together with VLR and the new fusion scheme, the proposed MPS, in general, can provide superior performance compared with the state-of-the-art works on four public datasets.**

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

## I. Introduction

With the advancement of miscellaneous sophisticated object detection techniques [1, 2, 3], action localization in videos, which lies at the core of applications such as self-driving cars and humanoid robots [4, 5, 6, 7], has become an emerging research topic. However, due to the complicated nature of occlusion, low-resolution videos, nonsteady viewpoints, *etc.*, automatic action localization, especially for multiple objects, has been a challenging task.

A myriad of algorithms has been addressed to effectively and efficiently localize single or multiple path actions in videos. For example, the temporal sliding window approach was employed in [8, 9, 10, 11] by using the handcrafted features for action classification. Gemert *et al.* [12] bypassed the segmentation process by generating proposals directly from the dense trajectory and used a single-link clustering method [13] to localize the action. Mettes *et al.* [14] utilized pointly-supervised proposals, an annotation scheme with a minimum amount of supervision, to speed up action location annotation. However, these approaches do not use the features generated by the potent convolutional neural network (CNN); thus, their performance is generally inferior to that of CNN-based methods. The spatio-temporal information was obtained using the CNN features in [6], and the bounding boxes were then linked together by a learning process using the SVM classifier. However, its complexity is not simple 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, some dynamic programming (DP)-based methods [4, 5, 15, 16, 17, 18] have been used to detect multiple action tubes in videos. For instance, Gkioxari *et al.* [5] used R-CNN as the object detector to generate the bounding boxes in both the spatial and motion domains. Saha *et al.* [4] employed faster R-CNN instead of R-CNN to further enhance the performance. Wang *et al.* [15] proposed a two-stream hybrid fully convolutional network for actionness estimation. Zolfaghari *et al.* [16] made use of a Markov chain model to successively combine cues in multi-stream networks. Peng *et al.* [17] modified the faster R-CNN model and included a multi-region scheme to improve the detection performance. Yang *et al.* [18] attempted to increase the localization accuracy and temporal consistency by utilizing cascade region proposal and location anticipation networks. All of these algorithms [4, 5, 15, 16, 17, 18], however, use the standard DP, which does not fully utilize the information generated in each iteration and has to restart the search to find one action path after another for multiple path scenarios, thus entailing high complexity, especially for videos with many instances or frames.

This paper presents an effective two-stream faster R-CNN-based action localization approach to detect multiple spatial-temporal action tubes in videos. Since the inaccurate bounding boxes produced by the networks may impact the detection accuracy, a video localization refinement (VLR) scheme is first addressed to iteratively rectify the potentially inaccurate bounding boxes by exploiting the temporal consistency between adjacent frames. Thereafter, to provide more faithful detection scores, a new fusion strategy is considered, which combines not only the appearance and the flow information of the two-stream R-CNN but also the motion saliency, the latter of which is included to address the small camera motion. Moreover, an efficient DP-like algorithm, termed multiple path search (MPS), is developed to find multiple paths in a single run. MPS comprises two passes: forward message passing and backward path tracing. In the forward message passing, each node stores the information of a prescribed number of connections based on the accumulated scores determined in the previous stages instead of only the largest one in DP. A

Erick Hendra Putra Alwando, Yie-Tarng Chen, and Wen-Hsien Fang are with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology Taipei, Taiwan, R.O.C. Email: {m10402805, ytchen, whf}@mail.ntust.edu.tw

backward path tracing is invoked afterward 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. Simulation results show that together with VLR and the new fusion strategy, the proposed MPS algorithm, in general, can provide superior performance compared with the state-of-the-art works on four publicly available datasets.

The contributions of this paper include the following: 1) a novel low-cost DP-like algorithm is developed to efficiently find *multiple* paths in a single run; 2) a new fusion strategy, which is resilient to small camera motion, is addressed by combining the appearance and the flow information of the two-stream faster R-CNN along with motion saliency; and 3) an iterative VLR scheme, which can rectify the potentially inaccurate bounding boxes generated by the two-stream networks into their correct position and scales, is addressed to further enhance the detection accuracy. Part of this paper has been presented at the IEEE International Conference on Image Processing (ICIP) 2017 [19].

The rest of this paper is organized as follows. Sec. II gives an overview of the related work. Sec. III introduces the proposed multiple path search method, which includes a new video localization refinement scheme, a new fusion strategy and a multiple path search approach. Extensive experiments are conducted in Sec. IV to assess the performance of the new method. Sec. V draws some concluding remarks to conclude our work.

## II. Related Work

A considerable amount of research on object proposal generation [20, 21, 22, 23, 24] has been investigated for action localization. For example, object detection [5, 6] used unsupervised region proposal algorithms such as selective search [25] and edge boxes [26]. However, these approaches are very time-consuming due to the multi-stage classification scheme which requires CNN fine-tuning at both the training and testing stages. Gemert *et al.* [12] utilized unsupervised clustering to create a small set of spatio-temporal region proposals. However, since [12] uses dense-trajectory features [27], it does not work well for actions with small motion.

CNN-based object detection can be classified into two categories: region proposal object detectors and proposal-free object detectors. Our approach belongs to the former, for which it is required to generate object proposals before object detection. The proposed MPS and VLR are both based on the region proposal object detectors where the detection accuracy is a major concern. For instance, R-CNN [28] and spp-net [29] used the Zeiler and Fergus's (ZF) CNN architecture [30] as a feature extractor. [1, 31] replaced the ZF CNN architecture with the very deep CNN (VGG-16) architecture [32] and used the multitask loss in a single CNN network to further boost the performance. Donahue *et al.* [33] showed that utilization of semantic deep features based on [34] can outperform many visual recognition tasks, such as object recognition, fine-grained part-based, and event detection. Recently, He *et al.* [35] extended the work [1] for instance segmentation by adding a mask head to classify the region inside the detected bounding box and replacing the region-of-interest (RoI) pooling layer with a more accurate RoI align layer.

The second category is the proposal-free object detectors which do not need to generate object proposals to identify the classified bounding boxes and directly predict the bounding boxes and confidence of each category by a single neural network. Wang *et al.* [15] directly transformed raw images into an actionness map using fully convolutional networks, but the complexity of which is not simple. Redmon *et al.* proposed a real-time CNN-based object detector, YOLO [3]. Another popular proposal-free object detector is the single shot multibox detector (SSD) [2], which increased the detection accuracy by predicting the category scores and box offsets for a fixed set of default boxes, similar to *anchors* in [1], using multiple scales of the feature map. Although these algorithms can provide competing performance with low complexity, they generally struggle with inaccurate localization and do not work well in locating objects of small scales even though the image resolution has been raised [36]. Meanwhile, these methods do not generalize well to objects with a new aspect ratio.

As for temporal localization, the majority of works used the time-consuming sliding window approach [10, 37, 38]. For instance, Tian *et al.* [10] extended the deformable part models, introduced in [39] for object detection in 2D images, to 3D images by using HOG3D descriptors [40] and then employed a sliding window approach. Laptev *et al.* performed a sliding window on cuboids, thus restricting the action to a fixed spatial extent across frames. To overcome this setback, Oneata *et al.* [9] proposed a new temporal sliding strategy, which combined rescoring and non-maximum suppression (NMS) to reduce the complexity while maintaining the temporal consistency. Bharati *et al.* [41] made use of a kernelized correlation filter to keep track of the object detection in real time.

The recurrent neural network (RNN) or its variant long short-term memory (LSTM) have also been utilized extensively to obtain discriminative features for action recognition or action localization. For instance, Veeriah *et al.* [42] included the differential of states in conventional LSTM gates to model the changes in spatio-temporal information. Hu *et al.* [43] proposed another version of RNN that took the frame-level features from a CNN to encode the features of actions in the videos. These two works, however, focused on the action recognition problems, which only describe what actions are happening without specifying the locations of the actions in each frame. In contrast, this paper focuses on action localization, which localizes the actions within frames while maintaining the temporal consistency for each action through the construction of action tubes. Additionally, Li *et al.* [44] proposed a convolutional soft-attention LSTM to address the lack of spatial correlation information in the conventional LSTM when dealing with video input. Although RNN can handle periodic data effectively, we do not utilize it in our framework because such networks, in general, require a large amount and a variety of training data, and its performance degenerates when dealing with highly aperiodic action movement [45].

Action localization depends heavily on the quality of the

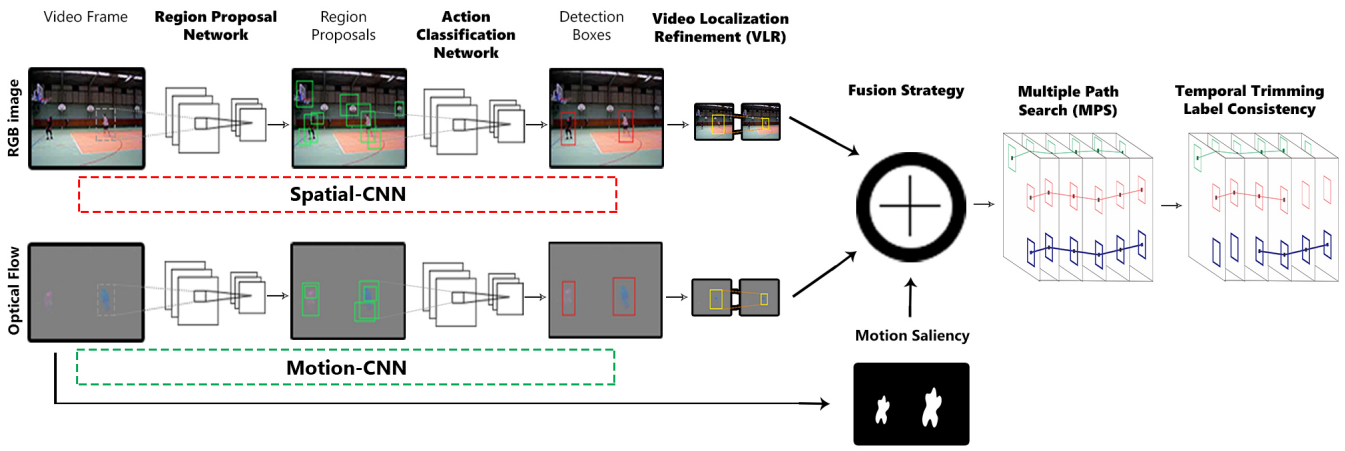


Fig. 1: Overview of the proposed action localization framework.

image features. To account for this, Simonyan *et al.* [46] proposed a two-stream CNN-based approach to address the action recognition problem by concatenating the appearance information along with the motion information from two independent networks. Feichtenhofer *et al.* [47] made use of motion gating as the fusion strategy in the two-stream architecture. Redmon *et al.* [3] combined the output of the fast R-CNN with lower localization error and that of the YOLO object detection networks with lower background error to improve the detection accuracy. Dai *et al.* [48] employed a temporal context representation for ranking proposals to obtain high recall. Saha *et al.* [4], inspired by [3], considered a new fusion strategy that combined the detection score of the appearance bounding box with that of the motion bounding box if these two bounding boxes overlap with each other.

A number of methods have been addressed for the refinement of action proposals [49, 28, 50]. For instance, Szegedy *et al.* [49] addressed object proposal refinement by combining more than one object proposal [25, 51] to obtain as many accurate region proposals as possible. However, this approach must address many highly overlapped region proposals, which in turn incur high complexity. Sermanet *et al.* [50] refined the sliding window based region proposals using a regression network that mapped the features of the last convolutional layer to refine the box. Girshick *et al.* [28] predicted a new bounding box using a linear regression model. However, a retraining process is necessary for those methods when applied to other datasets. Cheng *et al.* [52] addressed a training-free iterative process to refine the bounding box. Similarly, Gidaris *et al.* [53] considered an iterative scheme to enhance the location accuracy of the bounding boxes, but with a different probability model. These approaches [52, 53], however, were only devised for images.

## III. The Proposed Method

In this section, we begin with the introduction of the overall structure of the proposed action localization method in Sec. III-A. Our focus is then on three main steps. First, to rectify the possibly inaccurate bounding boxes, a new video localization refinement scheme is described in Sec. III-B. Next, Sec. III-C addresses an effective fusion strategy to obtain more faithful detection scores. Finally, an efficient multiple path search algorithm is addressed in Sec. III-D.

### A. Overall Methodology

Our two-stream R-CNN action localization method consists of six steps, the first two of which are the same as faster R-CNN [1], while the last one follows [4]. The third to the fifth steps are the new ones to be delineated in the succeeding subsections. For easy reference, the overall procedures of the proposed method are as illustrated in Fig. 1.

**Region Proposal Network (RPN):** First, region proposals are generated [1] to facilitate action localization in later stages. This stage extracts a set of candidate regions at the frame level from either the RGB images or the optical flow images. These CNN-based networks are trained to discriminate the action from its background. All of the bounding boxes generated in this stage do not contain any action. Later, in the action classification network, each of the region proposals will be classified by the CNN detector.

**Action Classification Network:** Second, each of the generated region proposals is extracted using a fast R-CNN [1] model to obtain the deep features, and a softmax score is determined for each specific action class in the last layer of the CNNs, which uses VGG-16 as the base architecture in the faster R-CNN object detector framework. After this stage, every region proposal has $C$ multiclass softmax action scores, where $C$ is the number of action classes in the dataset used to train the detection model.

**Video Localization Refinement:** Third, VLR is invoked for both the spatial and flow domains of the two-stream faster R-CNN to iteratively refine the potentially inaccurate bounding boxes.

**Fusion Strategy:** Fourth, we consider a new fusion strategy that leverages the appearance and the flow information of the two-stream R-CNN and the motion saliency. Motion saliency

is taken into account to complement the use of the optical flow information for some scenarios that involve small camera motion. The score determined in this stage is our action detection score and will be used in the succeeding two stages.

**Multiple Path Search:** Fifth, a new DP-like algorithm, MPS, is proposed, which makes full use of the information in the forward message passing to simultaneously find all paths in the backward path tracing within one iteration.

**Temporal Trimming Label Consistency:** Finally, based on [4], temporal label trimming consistency is invoked to differentiate the action labels from the background in the temporal domain by analyzing the action path scores using DP. If a sequence of scores is consistent and smooth, then it is categorized as an action.

### *B. Video Localization Refinement*

To rectify the detection boxes generated by the action classification network into their correct positions and scales, we consider a VLR method, which is an extension of [52] from images to videos by exploiting the temporal consistency between adjacent frames. Let $\mathbf{x}_{i,j}$ denote the detection box for the $j^{th}$ object from either the spatial or flow domains generated by faster R-CNN in frame $i$ with a detection score $s_c(\mathbf{x}_{i,j})$ for a particular class $c$. Every bounding box $\mathbf{x}_{i,j}$ is parameterized with four-dimensional vectors $[t, b, l, r]$, which correspond to the top, bottom, left and right, respectively. In each iteration, VLR first finds a new bounding box with the largest detection score in the search region. Then, inspired by the mean shift algorithm [54], we shift the center of the search region to the new bounding box and repeat the iteration until convergence. The overall VLR, as shown in Fig. 2, is comprised of the following two steps in each iteration:

**Determination of the search region:**

We begin by determining the search region $\mathcal{X}_{i,j}$ for the input detection box $\mathbf{x}_{i,j}$ by considering the region proposals from 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 $\mathbf{x}_{i,j}$, where $\gamma \in [0, 1]$ is the degree of overlap and the intersection-over-union function (IoU) between two bounding boxes, $\mathbf{x}_1$ and $\mathbf{x}_2$, is defined as [55]

$$\mathsf{IoU}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathsf{area}(\mathbf{x}_1 \cap \mathbf{x}_2)}{\mathsf{area}(\mathbf{x}_1 \cup \mathbf{x}_2)}. \tag{1}$$

The search region $\mathcal{X}_{i,j}$ for the 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 $\mathbf{z}_\cup = \cup_{\mathbf{z} \in \mathcal{N}(\mathbf{x}_{i,j})} \mathbf{z}$. Fig. 2(a) shows an example of the video localization refinement method, where the yellow rectangle is the input bounding box $\mathbf{x}_{i,j}$, and the black rectangles are the region proposals, which are the elements of $\mathcal{N}(\mathbf{x}_{i,j})$. As shown in Fig. 2(b), the green region bounded by the white dotted rectangle is the search region $\mathcal{X}_{i,j}$, which is determined by the smallest and the largest bounding boxes in $\mathcal{N}(\mathbf{x}_{i,j})$.

**Localization refinement:**

Next, we attempt to find a new bounding box $\mathbf{r}_{i,j}$ in the search region $\hat{\mathcal{X}}^*_{i,j}$ with the highest detection score, where $\mathcal{X}^*_{i,j} = \{\mathbf{y} | \mathbf{y} \in \hat{\mathcal{X}}_{i,j}, s_c(\mathbf{y}) > s_c(\mathbf{x}_{i,j})\}$ denotes a set of bounding boxes in $\hat{\mathcal{X}}_{i,j}$ with scores larger than $s_c(\mathbf{x}_{i,j})$ and $\hat{\mathcal{X}}_{i,j}$, a set of bounding boxes uniformly sampled in the search region $\mathcal{X}_{i,j}$, is given by

$$\hat{\mathcal{X}}_{i,j} = \hat{T}_{i,j} \times \hat{B}_{i,j} \times \hat{L}_{i,j} \times \hat{R}_{i,j}, \tag{2}$$

where $\hat{T}_{i,j} \times \hat{B}_{i,j} \times \hat{L}_{i,j} \times \hat{R}_{i,j}$ are sets of integers containing the evenly spaced points in the intervals $[t_\cup, t_\cap], [b_\cap, b_\cup], [l_\cup, l_\cap], [r_\cap, r_\cup]$, as shown in Fig. 3. For efficiency, we limit all $|\hat{T}_{i,j}|$, $|\hat{B}_{i,j}|$, $|\hat{L}_{i,j}|$, and $|\hat{R}_{i,j}|$ to be less than or equal to 5, where $|\mathcal{A}|$ denotes the cardinality of the set $\mathcal{A}$. To determine whether the iterations should proceed, we consider an indicator function associated with the subset $\mathcal{X}^*_{i,j}$ as $\mathcal{L}(\mathcal{X}^*_{i,j}) = |\mathcal{X}^*_{i,j}|/|\mathcal{X}_{i,j}|$, which estimates how likely it is that the desired bounding box $\mathbf{r}_{i,j}$ exists inside $\mathcal{X}^*_{i,j}$. If $\mathcal{L}(\mathcal{X}^*_{i,j}) > 0$, which implies that the subset $\mathcal{X}^*_{i,j} \neq \emptyset$, then we can find a new bounding box $\mathbf{r}_{i,j}$ with the largest detection score by an exhaustive search over $\mathcal{X}^*_{i,j}$. Thereafter, we shift the center of the search region from $\mathbf{x}_{i,j}$ to $\mathbf{r}_{i,j}$, as shown in Figs. 2(c) and (d) and repeat the above steps to further refine the bounding box, as shown in Fig. 2(e) until no better bounding box is available, *i.e.* $\mathcal{L}(\mathcal{X}^*_{i,j}) = 0$, or the number of iterations exceeds the prescribed $\mathsf{I}_0$. At the end of the iterative process, we can obtain the desired bounding box along with its detection score, as shown in Fig. 2(f).

For the convergence issue, based on the developed iterative scheme, the detection score of the optimal bounding box determined in each iteration will be larger than that in the previous iteration. Additionally, the optimum detection score determined in each iteration is always bounded above as the search region is finite. Consequently, these detection scores will form a monotonically increasing sequence, so the convergence of the proposed VLR is ensured because a bounded, monotonically increasing sequence will at least converge to a local maximum [56].

### *C. Fusion Strategy*

In this subsection, we consider a new fusion strategy based on the two-stream faster R-CNN. In contrast to previous works such as [4, 6, 15], which only employ the appearance information from the RGB images and the motion information from the optical flow images, inspired by [5], here we incorporate the motion saliency with the above to reduce the adverse camera motion effect to obtain more accurate detection scores. This consideration is because some false bounding boxes may be generated by the motion-based CNN incurred by small camera motion. With the enforcement of the motion saliency, the region containing the true moving object can be manifested to distinguish it from the false bounding boxes. As an illustration, for the image in Fig. 4 (a), based on the output of the motion-CNN, we can not differentiate the bounding box induced by the camera motion from the one by the target motion, as shown in Fig. 4(b), as the detection scores of both are similar. Together with the motion saliency in Fig. 4(c), we can reinforce the motion information of the target object. Note that we also keep the optical flow information

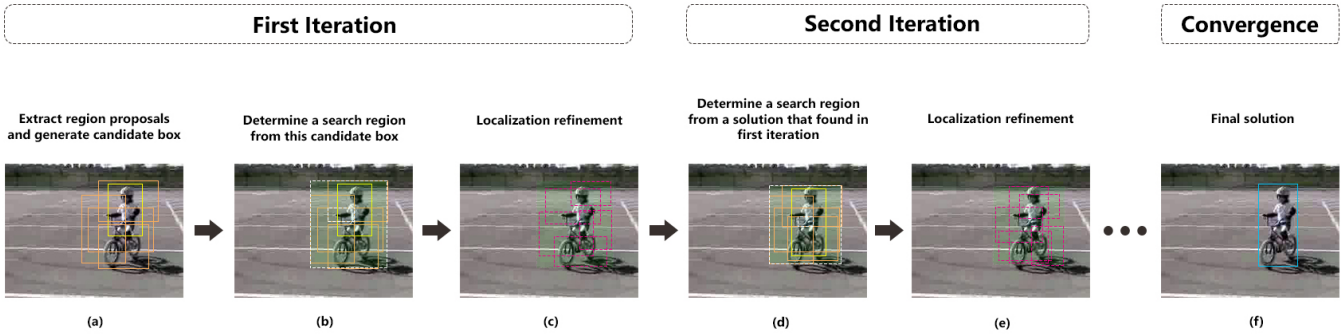


Fig. 2: Overview of the proposed video localization refinement: region proposals have orange bounding boxes; the detection box is marked with a yellow bounding box; the search regions are shaded with green and contoured with a white dotted bounding box, and the regions after fine-tuning are marked with red bounding boxes.

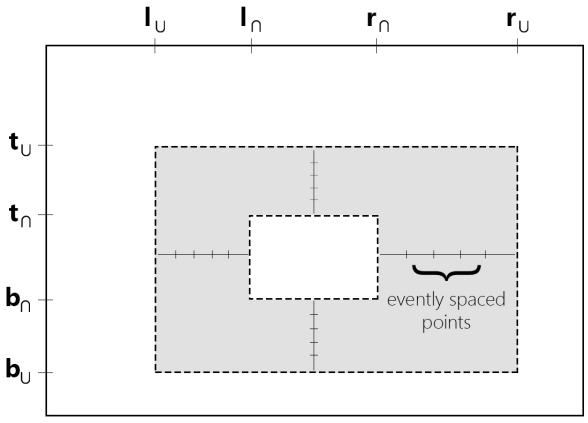


Fig. 3: Illustration of the search region $\hat{\mathcal{X}}_{i,j}$.

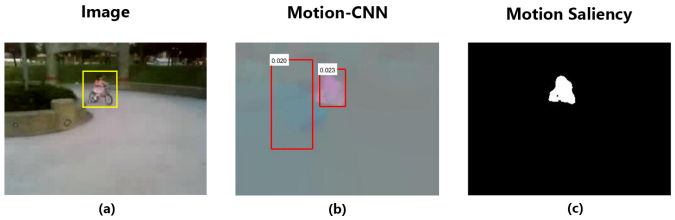


Fig. 4: Overview of the motion information (optical flow and motion saliency): (a) the target object is marked with the yellow bounding box; (b) detection boxes using motion-CNN (the left bounding box is due to the camera motion, and the right bounding box belongs to the target); (c) the object generated by motion saliency.

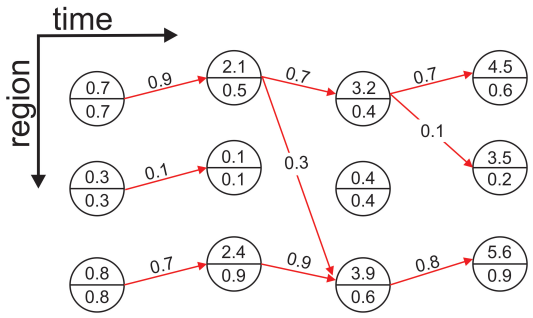


Fig. 5: A forward message passing example: A node denotes a region $\mathbf{r}_{i,j}$ with upper value $S_{i,j}$, lower value $s_c(\mathbf{r}_{i,j})$, and the value above the arrow denotes IoU. An arrow pointing from node A to B indicates that A in frame $(i-1)$ is the neighbor of B in frame $i$, and A further passes the message to B if this arrow is solid.

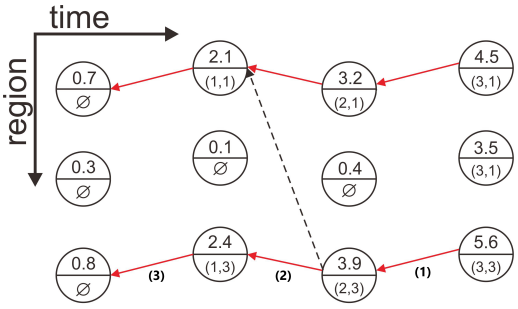


Fig. 6: A backward path tracing example: A path is traced back with solid arrows. The dotted arrow indicates a failed connection since the pointed node is used by the other path.

from the motion-CNN, as in some cases the motion saliency is insensitive to the objects with small motion.

Denote the refined detection boxes in the spatial and the 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 $s^*_c(\mathbf{r}_{i,j})$. Based on the new fusion strategy, the detection score is now a weighted summation of the detection scores of the spatial and flow bounding boxes and the motion saliency score given by

$$s^*_c(\mathbf{r}_{i,j}) = s_c(\mathbf{r}^s_{i,j}) + \beta \times s_c(\mathbf{r}^f_{i,max}) \times \mathrm{IoU}(\mathbf{r}^s_{i,j}, \mathbf{r}^f_{i,max}) + \eta \times w(\mathbf{r}^s_{i,j}), \quad (3)$$

where $\mathbf{r}^f_{i,max} = \mathrm{argmax}_{\{\mathbf{r}^f_{i,n} \ \forall \ n\}} \mathrm{IoU}(\mathbf{r}^s_{i,j}, \mathbf{r}^f_{i,n})$, $w(\mathbf{r}^s_{i,j})$ is the motion saliency score, and $\beta$ and $\eta$ are properly chosen weighting parameters. When $\beta = 1$ and $\eta = 0$, Eq. (3) will reduce to the fusion strategy in [4]. Following [5], we use

**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$\{A_c(\mathbf{r}_{i-1,k^j}, \mathbf{r}_{i,j}) | k^j = 1, .., N\}$
      and return $A_c(\mathbf{r}_{i-1,k_1^j}, \mathbf{r}_{i,j}) \geq \ldots$
      $\geq A_c(\mathbf{r}_{i-1,k_N^j}, \mathbf{r}_{i,j})$
      set $\mathbf{k}^j = \{k_1^j, \ldots, k_K^j\}$
      **if** $\mathbf{k}^j \neq \varnothing$ **then**
        set $S_{i,j} = S_{i-1,k_1^j} + A_c(\mathbf{r}_{i-1,k_1^j}, \mathbf{r}_{i,j})$
        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** $\mathbf{S}$,$\mathbf{P}$ (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**

the normalized magnitude of the optical flow signal $f_m$ as a heat map, which contains the pixels associated with the object motion. More specifically, $f_m(V) = \frac{1}{|V|}\sum_{v \in V} f_m(v)$ is a measure of how motion salient region $V$ is. For region $V$, if $f_m(V) \geq \delta$, where $\delta$ is a predetermined threshold, then $V$, referred to as a salient region, is retained; otherwise, it will be discarded. Let $R$ be a prescribed window that slides across frame $i$ and $H_i$ denotes the entire set of salient regions in this frame, and we then define the (normalized) motion saliency score for the bounding box $\mathbf{r}_{i,j}^s$ as

$$w(\mathbf{r}_{i,j}^s) = \frac{area(\mathbf{r}_{i,j}^s \cap H_i)}{area(H_i)}. \tag{4}$$

With the definition given in Eq. (4), we observe that if the bounding box $\mathbf{r}_{i,j}^s$ has a high motion saliency score, then it is more likely that it contains a moving object.

### D. Multiple Path Search Algorithm

To follow, we develop an efficient DP-like algorithm to simultaneously find all multiple paths. Denote a video path as $T$, which is a collection of refined detection boxes $\{\mathbf{r}_{i,j}\}$ together with their action detection scores $\{s_c^*(\mathbf{r}_{i,j})\}$ after the fusion described in Sec. III-C in each frame. These detection boxes can be linked up to determine the regions most likely to be associated with a single action instance, called an *action tube*. Our objective is to maximize the accumulative score given by

$$\sum_{T_1, \cdots, T_N} \sum_{i=1}^{M-1} A_c(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}), \tag{5}$$

where

$$A_c(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}) = s_c^*(\mathbf{r}_{i+1,l}) + \alpha \times \mathsf{IoU}(\mathbf{r}_{i,j}, \mathbf{r}_{i+1,l}), \tag{6}$$

in which the subscripts $j$ and $l$ are dictated by the paths $\{T_n\}_{n=1}^N$, $s_c^*(\mathbf{r}_{i,j})$ is the action detection score of the bounding box $\mathbf{r}_{i,j}$ in Eq. (3), $N$ is the number of paths, $M$ is the total number of frames and $\alpha$ is the weighting parameter. We determine the accumulative score in Eq. (5) for every action tube and select the one with the largest accumulated score as the predicted action.

The DP approach was addressed in [4, 5, 15, 16, 17, 18] to solve the multiple path detection problem. However, since the standard DP was devised to find the optimal single path, the extension to multiple path scenarios has to iteratively restart DP to find one path after another and thus is very inefficient. To effectively address the multiple path search problem, we propose a new DP-like approach as described in Algorithm 1, which is comprised of two stages: forward message passing and backward path tracing. The forward message passing, which takes the refined detection boxes $\mathbf{r}_{i,j}$ and their scores $s_c^*(\mathbf{r}_{i,j})$ in Eq. (1) as input, is described in the upper section of Algorithm 1. In the beginning, each of the detection boxes in the first frame will initiate a new path. Then, the message passing begins with the second frame until the last of the video frames. Following the pass, each node stores the node information, which contains the connections with the spatially-nearby bounding boxes with the $K$ largest accumulated scores determined in the previous stages based on Eq. (6) instead of only the largest one in the standard DP, where $K$ is a parameter leveraging the complexity and performance of MPS. At the end of the video sequence, 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, described in the bottom section of Algorithm 1, we take the matrices $\mathbf{S}$ and $\mathbf{P}$ as an input to trace back all possible connections of every path in the video. To this end, we first sort the matrix $\mathbf{S}$ in descending order, which returns a set of indices $\{(i_1, j_1), (i_2, j_2), \ldots, (i_{MN}, j_{MN})\}$. Based on these indices, we determine the start index to trace the matrix $\mathbf{P}$. During the tracing, the indices in $\mathbf{P}$, which correspond to the best path, are marked to prevent other paths from overlapping with this path. If it has already been assigned to any path, we begin to consider the node with the second largest score stored

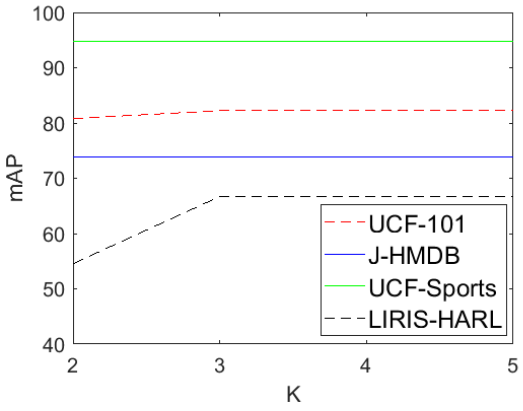


Fig. 7: Comparison of mAP versus $K$ using MPS.

in matrix $\mathbf{P}$. Finally, we flag each path as an action if its accumulated score exceeds the threshold.

As an illustration, an example of the forward message passing and the backward path tracing are shown in Figs. 5 and 6, respectively, where each edge represents the spatial-nearby score determined by Eq. (5). We note from Fig. 5 that the first node, which has $s_c^*(\mathbf{r}_{i,j}) = 0.7$, is connected to the node in the second frame with $s_c^*(\mathbf{r}_{i,j}) = 0.5$. The accumulated score, $S_{i,j}$, is obtained by adding both $s_c^*(\mathbf{r}_{i,j})$ from these two nodes along with their IoU based on Eq. (6). The accumulated score of each node is calculated only when the node in the adjacent frame has an IoU larger than a threshold. The process proceeds until the last video frame. In the backward path tracing shown in Fig. 6, we begin by selecting the node with the largest accumulated score (upper value) $S_{i,j} = 5.6$ in the last frame as the initial node and then sequentially choose the connection in every frame with the highest accumulated score and mark the chosen box. A toy sequence of these steps is marked with (1), (2), and (3) in the bottom of the arrows in Fig. 6.

## IV. Experimental Results

In this section, we first explain the details of the datasets employed in Sec. IV-A. Then, in Sec. IV-B, we describe the evaluation protocol and our experimental setup. Sec. IV-C evaluates the performance of the proposed MPS with various combinations of strategies and the impact of the number of connections $K$ stored in the nodes in the forward message passing of MPS. Afterward, we compare the proposed method with some state-of-the-art works in Sec. IV-D. Finally, the computational time of MPS is investigated in Sec. IV-E. All of the simulations are implemented in a computer with an Intel i7-6700 CPU, a 32 GB RAM, and GTX 1080 GPU.

### A. Datasets

In our experiments, we use four datasets: UCF-101, J-HMDB, UCF-Sports, and LIRIS-HARL. These four datasets have different characteristics to fully validate our new approach.

**UCF-101 dataset [57]:** This dataset contains 914 short untrimmed videos of various human actions and contains

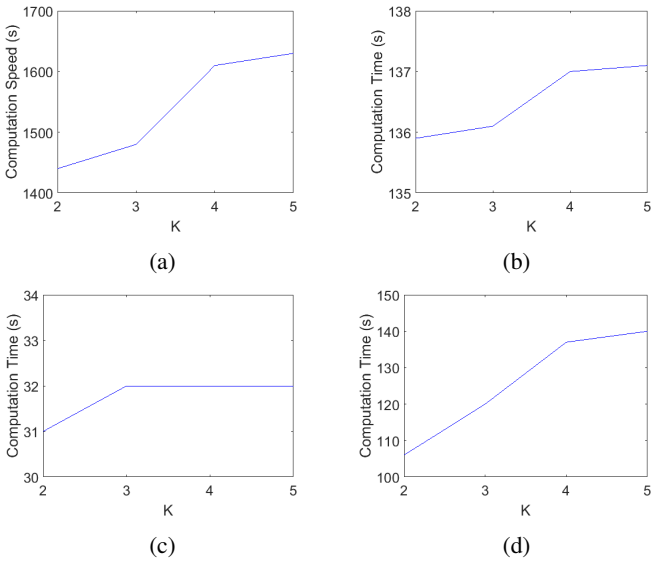


Fig. 8: Comparison of computation time versus K by MPS: (a) UCF-101 dataset. (b) J-HMDB dataset. (c) UCF-Sports dataset. (d) LIRIS-HARL dataset.

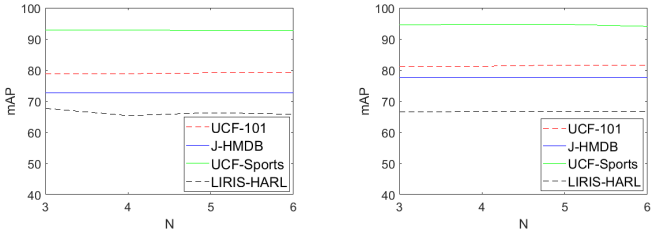


Fig. 9: Comparison of mAP versus $N$ using: (a) Standard DP. (b) MPS.

several viewpoints of actions, camera movements, clutters, and scales. It consists of 24 classes: basketball, basketball dunk, biking, cliff diving, cricket bowling, diving, fencing, floor gymnastics, golf swing, horse riding, ice dancing, long jump, pole vault, rope climbing, salsa spin, skate-boarding, skiing, jet skiing, soccer juggling, surfing, tennis swing, trampoline jumping, volleyball spiking, and walking with dog.

**J-HMDB dataset [58]:** This dataset is a subset of the HMDB dataset. Each video clip provided in this dataset is relatively short and between 15-40 frames. In total, it consists of 928 trimmed videos with 21 classes such as catch, clap, brush hair, jump, and swing baseball. Note that some video clips of this dataset still contain camera motion.

**UCF-Sports dataset [59]:** This dataset consists of 150 short videos from 10 different sports classes such as lifting and running. There is high intraclass similarity among the videos. All videos are trimmed to the action and the bounding box annotation is available for each frame. Most of the videos in the dataset contain lateral camera motion.

**LIRIS-HARL dataset [60]:** This dataset contains 167 untrimmed videos from 10 different action classes. This dataset is the most challenging compared to the other datasets.

TABLE I: Comparison of the action detection results (mAP) of MPS on the UCF-101 dataset (split 1) with various combinations of strategies. The best results are marked in bold.

| IoU threshold | | | | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
|---|---|---|---|---|---|---|---|---|---|---|
| Strategy | bounding box voting | new fusion strategy | VLR | | | | | | | |
| MPS | | | | 79.64 | 76.40 | 66.18 | 55.26 | 45.82 | 37.08 | 29.01 |
| MPS | ✓ | | | 79.01 | 76.17 | 66.41 | 56.66 | 47.09 | 37.49 | 29.05 |
| MPS | ✓ | ✓ | | 82.44 | 79.52 | 69.61 | 59.57 | 49.95 | 39.77 | 30.75 |
| MPS | ✓ | ✓ | ✓ | **85.79** | **82.41** | **72.90** | **62.32** | **52.36** | **41.10** | **32.10** |

TABLE II: Comparison of the action detection results (mAP) of MPS on the J-HMDB dataset (all splits) with various combinations of strategies. The best results are marked in bold.

| IoU threshold | | | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|---|---|---|---|---|
| Strategy | bounding box voting | new fusion strategy | VLR | | | | | | | |
| MPS | | | | 72.32 | 72.32 | 72.29 | 71.77 | 71.39 | 68.30 | 56.70 |
| MPS | ✓ | | | 72.65 | 72.64 | 72.60 | 72.14 | 71.38 | 68.53 | 56.49 |
| MPS | ✓ | ✓ | | 73.95 | 73.95 | 73.86 | 73.57 | 72.75 | 69.93 | 57.30 |
| MPS | ✓ | ✓ | ✓ | **79.84** | **79.78** | **79.76** | **79.57** | **78.26** | **74.78** | **58.48** |

TABLE III: Comparison of the action detection results (mAP) of MPS on the UCF-Sports dataset with various combinations of strategies. The best results are marked in bold.

| IoU threshold | | | | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.75 | 0.5:0.95 |
|---|---|---|---|---|---|---|---|---|---|---|
| Strategy | bounding box voting | new fusion strategy | VLR | | | | | | | |
| MPS | | | | 90.14 | 90.14 | 90.14 | 87.52 | 83.13 | 81.10 | 62.58 |
| MPS | ✓ | | | 90.14 | 90.14 | 90.14 | 87.52 | 83.13 | 81.10 | 63.44 |
| MPS | ✓ | ✓ | | 94.71 | 94.71 | 94.71 | 94.71 | 91.65 | 89.57 | 67.15 |
| MPS | ✓ | ✓ | ✓ | **94.71** | **94.71** | **94.71** | **94.71** | **91.65** | **89.57** | **67.48** |

TABLE IV: Comparison of the action detection results (mAP) of MPS on the LIRIS-HARL dataset with various combinations of strategies. The best results are marked in bold.

| IoU threshold | | | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|---|---|---|---|---|
| Strategy | bounding box voting | new fusion strategy | VLR | | | | | | | |
| MPS | | | | 59.06 | 50.33 | 42.75 | 33.77 | 27.76 | 23.95 | 8.22 |
| MPS | ✓ | | | 58.42 | 50.05 | 42.62 | 34.19 | 27.89 | 24.04 | 8.98 |
| MPS | ✓ | ✓ | | 65.55 | 59.85 | 47.24 | 40.85 | 34.72 | 25.51 | 10.99 |
| MPS | ✓ | ✓ | ✓ | **74.07** | **68.13** | **56.76** | **42.33** | **35.05** | **26.70** | **12.33** |

In addition to background clutter, more complicated occlusion, and several different viewpoints of actions, this dataset presents several additional difficulties since several actions share similar spatial and motion information but have different contexts, such as entering a room, trying to enter a locked room and unlocking a room, and some actions contain different appearances but with the same action, such as to put or to take something from a box. Furthermore, more than one action from the same or different classes can appear simultaneously in one video.

### *B. Evaluation Protocol and Experimental Setup*

We use the same architecture and training strategy as [1, 4] for both datasets by using VGG-16 with 5 convolutional layers interleaved with pooling and normalization and 3 fully connected layers. The training procedure for the spatial-CNN and motion-CNN are similar, except that the corresponding RPN and fast R-CNN are trained independently without sharing features. The spatial-CNN is trained using the RGB images, which are horizontally flipped and rescaled beforehand such that their shorter side is 600 pixels. The motion-CNN is trained using the single frame optical flow images [61]. The flows in the x- and y-directions and their magnitude are stacked to form a 3-dimensional image. As the RGB images, the optical flow images are also horizontally flipped and re-scaled. The training is performed with a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005 for 320 K iterations for the UCF-101 dataset, and 180 K iterations for the other datasets.

The simulations mainly follow the protocols and evaluation metrics provided by the UCF-101, the J-HMDB, and the LIRIS-HARL datasets [4, 57, 58, 60]. For the UCF-Sports dataset, we use the same train/test splits and evaluation protocol as [17, 59, 62]. For a fair comparison, we only use the UCF-101 dataset split 1 as the test set to obtain the final detection performance, whereas the reported results on the J-HMDB dataset are averaged over all three splits of the J-HMDB dataset. To determine the motion saliency score, we use a $5 \times 5$ spatial window $R$. The parameters $\beta$ and $\eta$ in Eq. (3), and $\alpha$ in Eq. (6) are determined using the grid search with cross-validation to attain the optimal performance. For the proposed VLR, we use $\gamma$=0.7 to control the degree of overlapping (IoU) in the determination of the search region and $\mathrm{l}_0 = 3$ for the maximum number of iterations. To remove the highly-overlapped bounding boxes in the adjacent regions after VLR, we invoke the standard NMS [28] with IoU = 0.3 to produce a smaller set of bounding boxes $\{\mathbf{r}_{i,j}^{nms}\}$ from those, $\{\mathbf{r}_{i,j}\}$, generated from VLR. Thereafter, the bounding box

voting scheme in [63] is employed, for which the coordinates of each bounding box in $\{\mathbf{r}_{i,j}^{nms}\}$ is adjusted by the bounding boxes in $\{\mathbf{r}_{i,j}\}$ with $\text{IoU}(\mathbf{r}_{i,j}, \mathbf{r}_{i,j}^{nms}) > 0.5$ using their scores $\{s_c^*(\mathbf{r}_{i,j})\}$ as the weights. The mean average precision (mAP) is utilized as the quantitative measure for performance evaluation. Additionally, because most of the videos in the UCF-101, the J-HMDB, the UCF-Sports, and the LIRIS-HARL datasets only consist of less than 5 actions, except for Figs. 9 and 14, the number of action paths $N$ is chosen as 5 in the simulations, as in [4, 5].

### C. Performance Evaluation

*1) Assessment of various combinations of strategies:* To follow, we scrutinize the performance improvement of MPS in terms of mAP with various strategies described above on the UCF-101, the J-HMDB, the UCF-Sports, and the LIRIS HARL datasets, as shown in Tables I to IV, respectively.

• **Impact of the bounding box voting:** First, we assess the impact of the bounding box voting scheme [63]. We note from Tables I to IV that this scheme can increase the mAP by approximately 0.1% to 0.6% for most of the IoUs. This is because the bounding box voting scheme reuses the highly concentrated bounding boxes to refine the bounding box coordinates.

• **Impact of the new fusion strategy:** Next, in addition to the bounding box voting, we consider the impact of the performance with the incorporation of our fusion strategy described in Sec. III-C, which leverages the appearance and the flow information of the two-stream faster R-CNN and the motion saliency. We can see from Tables I to IV that this new fusion strategy consistently improves mAPs by approximately 1% to 6% for most of the IoUs compared with the ones using [4]. This is mainly because motion saliency provides additional information to help us reduce the impact incurred by the small camera motion from the optical flow images. The improvement of the detection accuracy on the J-HMDB dataset is not as significant as that on the UCF-101 and UCF-Sports datasets because the former has limited and less shaky camera movement than the latter. Although there is no camera motion in the LIRIS-HARL dataset, the new fusion strategy still helps to enhance the detection by increasing the detection score of the target object. The improvement for the LIRIS-HARL dataset is more significant than the J-HMDB dataset, as the objects in the former are more difficult to localize than those in the latter.

• **Impact of VLR:** Finally, we inspect the performance improvement with the use of VLR. As shown in Tables I, II, and IV, apart from the above two schemes, when combined with VLR, the performance of MPS can be further increased by approximately 2% to 3%, 5% to 6%, and 2% to 9% on the UCF-101, the J-HMDB, and the LIRIS-HARL datasets, respectively. This is because that VLR can rectify the bounding boxes generated by the original faster R-CNN to those with more accurate positions, sizes, or scales, which in turn improve the action localization accuracy in the succeeding stage. We also note from Table I that there is only minor improvement on the UCF-Sports dataset. This is because the ground truth is loose, so the detection does not benefit from the bounding box refinement by VLR.

TABLE V: Comparison of the action detection results (mAP) on the UCF-101 dataset (split 1) with different IoUs. The best results are marked in bold.

| IoU threshold | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
|---|---|---|---|---|---|---|---|
| Mettes *et al.* [14] | - | - | 32.40 | - | - | - | |
| FAP [7] | 42.80 | - | - | - | - | - | - |
| Gemert *et al.* [12] | 58.00 | - | 37.80 | - | - | - | - |
| STMH [6] | 54.28 | 51.68 | 46.77 | 37.82 | - | - | - |
| VideoLSTM [44] | 70.00 | 54.00 | 38.00 | 20.00 | 7.00 | - | - |
| T-CNN [64] | 54.70 | 51.30 | 47.10 | 39.20 | - | - | - |
| Zolfaghari *et al.* [16] | 65.22 | 59.52 | 47.61 | 38.00 | - | - | - |
| Saha *et al.* [4] | 79.12 | 76.57 | 66.75 | 55.46 | 46.35 | 35.86 | 26.79 |
| MR-TS R-CNN [17] | 78.76 | 77.31 | 72.86 | **65.70** | **56.04** | 35.91 | 14.69 |
| CPLA [18] | 79.03 | 77.34 | **73.54** | 60.76 | 49.23 | 37.80 | - |
| Ours | **85.79** | **82.41** | 72.90 | 62.32 | 52.36 | **41.10** | **32.10** |

*2) Assessment of the Parameters $K$ and $N$:* Next, we investigate the choice of $K$ on the detection performance with IoU=0.05, where $K$ denotes the number of connections stored in each node in the forward message passing and depends on the number of actions in the videos. We compare the mAP performance of MPS with the bounding box voting and the new fusion schemes versus $K$ on the UCF-101, the J-HMDB, the UCF-Sports, and the LIRIS-HARL datasets, as shown in Fig. 7, from which we can see that the mAP performance is approximately the same after $K = 3$ for all datasets. This is because most of the videos in these datasets only consist of two actions on average at any time instant, so the information of three connections between the adjacent frames is sufficient. We also compare the computational time of this new algorithm versus $K$ on all datasets, as shown in Fig. 8. The computational time is the total time required by the algorithm to obtain all possible paths from all of the video frames in each dataset. Only the steps up to the fusion strategy is run with GPU and the other steps including MPS and the temporal trimming algorithm are only with CPU. We can see that the computational time of MPS only increases slightly with K because the overhead in determining and storing this extra connection information is much less than that required by DP. The simulations for other IoUs yield similar results. Based on this observation, we use $K = 3$ for all datasets hereafter.

We also scrutinize the impact of $N$ on the detection accuracy. We compare mAP versus $N$, as shown in Fig. 9, from which we can see that mAP is approximately the same for all $N$s when using either MPS or the standard DP. This is because in most cases, there are less than four actions at any time instant for all the datasets. After using NMS to pick the bounding boxes with low overlaps, the false detection is likely with a low detection score. Therefore, even if $N$ increases, both MPS and the standard DP will choose the path of the bounding boxes that have large accumulated scores.

### D. Comparisons with the State-of-the-Art Methods

In this subsection, we compare the proposed approach, which includes MPS along with the bounding box voting, the new fusion strategy, and VLR, with some recently reported works in terms of mAPs for different IoUs. The comparison

TABLE VI: Comparison of the action detection results (mAP) on the J-HMDB dataset with different IoUs. The best results are marked in bold.

| IoU threshold | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|---|---|
| ActionTube [5] | - | - | - | - | 53.30 | - | - |
| Wang *et al.* [15] | - | - | - | - | 56.40 | - | - |
| STMH [6] | - | 63.10 | 63.50 | 62.20 | 60.70 | - | - |
| Saha *et al.* [4] | 72.65 | 72.63 | 72.59 | 72.24 | 71.50 | 68.73 | 56.57 |
| MR-TS R-CNN [17] | - | 74.30 | - | - | 73.10 | - | - |
| Zolfaghari *et al.* [16] | 78.81 | 78.20 | 77.12 | 75.05 | 73.47 | - | - |
| T-CNN [64] | - | 78.40 | - | - | 76.90 | - | - |
| Ours | **79.84** | **79.78** | **79.76** | **79.57** | **78.26** | **74.78** | **58.48** |

TABLE VII: Comparison of the action detection results (mAP) on the UCF-Sports dataset with different IoUs. The best results are marked in bold.

| IoU threshold | 0.2 | 0.5 | 0.6 | 0.7 | 0.75 | 0.5:0.95 |
|---|---|---|---|---|---|---|
| ActionTube [5] | - | 75.80 | - | - | - | - |
| STMH [6] | - | 90.50 | - | - | - | - |
| MR-TS R-CNN [17] | 95.12 | 95.12 | 90.34 | 73.70 | 47.33 | 50.92 |
| T-CNN [64] | **95.20** | **95.20** | - | - | - | - |
| Ours | 94.71 | 94.71 | **94.71** | **91.65** | **89.57** | **67.50** |

results with ten baselines, Mettes *et al.* [14], FAP [7], Gemert *et al.* [12], STMH [6], VideoLSTM [44], Zolfaghari *et al.* [16], Saha *et al.* [4], T-CNN [64], MR-TS R-CNN [17], and CPLA [18] on the UCF-101 dataset are shown in Table V, from which we can see that [12, 14] do not provide satisfactory performance. This is mainly because [12, 14] focus on the speedup of the whole process by generating proposals directly from the dense trajectory and annotating them with a minimum amount of supervision without using the CNN architecture. With only action labels, VideoLSTM [44] has better localization accuracy than [14, 7, 12, 6, 64] for IoU $\leq$ 0.1. However, its performance drops quickly for larger IoUs due to the inferior spatial localization which relies only on saliency maps and greedy search. Additionally, it is not devised for multiple path search scenarios.

We can also see that [4] outperforms [14, 7, 12, 6, 44, 64, 16] for all IoUs, as [4] uses faster R-CNN to obtain the detection boxes and the potent DP algorithm to find multiple paths. We can see that [17] and [18] achieve high detection accuracy for small IoUs because both of them use temporally aligned spatio-temporal information to boost the consistency of action localization in the temporal domain. The proposed algorithm provides the best performance compared with the aforementioned works for most of the IoUs by incorporating MPS with a new fusion strategy that contains motion saliency information to reduce the impact of slow camera motion and VLR, which can iteratively rectify the bounding box coordinates to attain more precise ones. In particular, the performance of the new approach does not drop quickly like [17] when IoU becomes large.

The performance comparison is also made on the J-HMDB dataset with seven baselines, including ActionTube [5], Wang *et al.* [15], STMH [6], Saha *et al.* [4], MS-TS R-CNN [17], Zolfaghari *et al.* [16], and T-CNN [64], as shown in Table VI. We see in Table VI that [5], [15], and STMH [6] can not provide satisfactory detection accuracy, as these methods do not employ the region proposal networks to generate more

TABLE VIII: Comparison of the action detection results (mAP) on the LIRIS-HARL dataset with different IoUs. The best results are marked in bold.

| IoU threshold | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|
| Saha *et al.* [4] | 54.18 | 49.10 | 35.91 | 28.03 | 21.36 |
| CPLA [18] | - | 54.34 | - | - | - |
| Ours | **74.07** | **68.13** | **56.76** | **42.33** | **35.05** |

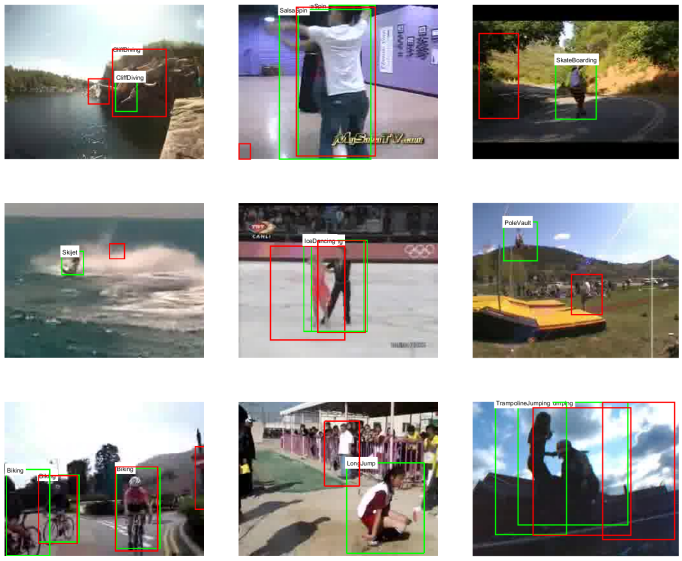

Fig. 10: Some action detection results by our algorithm and [4], in green and red, on the UCF-101 dataset.

precise object proposals. Zolfaghari *et al.* [16] addressed the overfitting problem in their fusion strategy by using the Markov decision model to render significant performance improvement compared with [4] and [17], which perform fusion based only on the detection scores. The T-CNN [64] has a slight improvement over [16] by incorporating a 3D convolutional neural network to perform multitask spatio-temporal detection. We also find that the proposed algorithm consistently outperforms all of the other works and has approximately 5% and 1.5% gains over [16] and [64], respectively, when IoU = 0.5. This is because [16] only uses body part segmentation instead of the more accurate faster R-CNN to produce the bounding boxes, while joint learning in [64] is difficult to optimize for such a complex dataset. The gains become more significant when dealing with longer and more challenging videos such as the UCF-101 dataset.

Next, we compare the detection performance of our method on the UCF-Sports dataset with four baselines, ActionTube [5], STMH [6], MR-TS R-CNN [17], and T-CNN [64], as shown in Table VII, from which we see that the performance on this dataset is superior over the previous two datasets, as the actions in each class are similar and thus are easier to differentiate. Additionally, MR-TS R-CNN [17] outperforms [5, 6] by utilizing a more effective two-stream network with stacked optical flow features. Our method has approximately the same performance as [17, 64] at lower IoUs and can

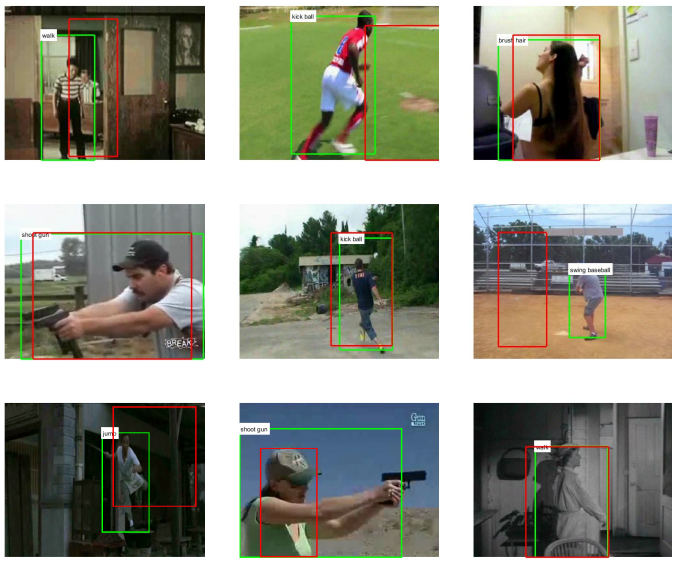

Fig. 11: Some action detection results by our algorithm and [4], in green and red, on the J-HMDB dataset.

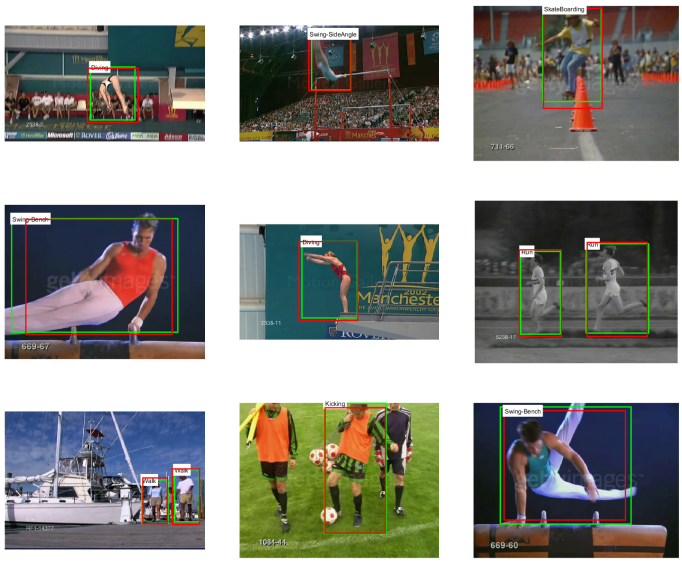

Fig. 12: Some action detection results by our algorithm and [4], in green and red, on the UCF-Sports dataset.

achieve the best performance through the use of the new fusion strategy that takes account of the motion saliency to manifest the motion information.

Finally, we compare the LIRIS-HARL dataset with two baselines, Saha *et al.* [4] and CPLA [18], as shown in Table VIII, from which we can see that the detection performance of [4] on this dataset is inferior to that on the previous datasets. This is because the actions in this dataset are more involved and require more high-level features to capture the semantics and context of the actions. Moreover, most of the actions only occupy a short time interval in the videos. We find that CPLA [18] provides better performance than [4] because it leverages

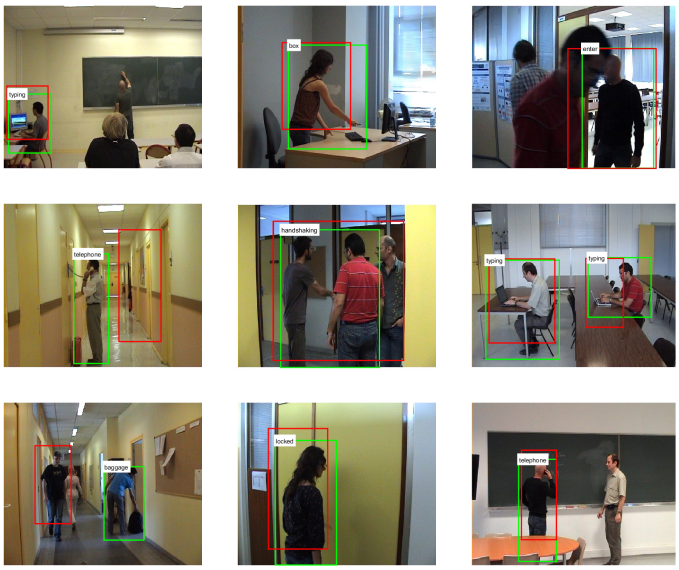

Fig. 13: Some action detection results by our algorithm and [4], in green and red, respectively, on the LIRIS-HARL dataset.

TABLE IX: Comparison of the search speed (frame per second, fps) on the UCF-101, the J-HMDB, the UCF-Sports, and the LIRIS-HARL datasets.

| Method | UCF-101 | J-HMDB | UCF-Sports | LIRIS-HARL |
|---|---|---|---|---|
| Standard DP | 72 | 58 | 67 | 60 |
| MPS | 101 | 66 | 77 | 97 |

the proposal quality with a cascaded proposal network and strengthens the temporal consistency by predicting the action movement through the local anticipation network. Our method outperforms [18] because it can effectively enforce the consistency of the detection with the help of more accurate bounding boxes by VLR and the new fusion strategy.

As an illustration, some detection results of our algorithm, which comprises MPS with all of the aforementioned strategies and [4] on all datasets, are also furnished in Figs. 10 to 13, from which we can see that the detection boxes based on the proposed algorithm are more accurate compared with those based on [4]. In some cases, our algorithm can detect the object, while [4] cannot. This is because sometimes VLR with the new fusion scheme can correct false positive bounding boxes into true positive bounding boxes.

### *E. Computation Time Analysis*

To assess the complexity, we compare the search speed of the proposed MPS and the standard DP with IoU=0.05 based on the UCF-101, the J-HMDB, the UCF-Sports, and the LIRIS-HARL datasets as shown in Table IX. Note that [4, 5, 15, 16, 17, 18] require approximately the same time complexity on both datasets, as they are all based on the standard DP algorithm. The speed of VLR is approximately 3 frames per second. From Table IX we can see that MPS is approximately 30% faster than the standard DP on the UCF-101 and the LIRIS-HARL datasets. This is because MPS only

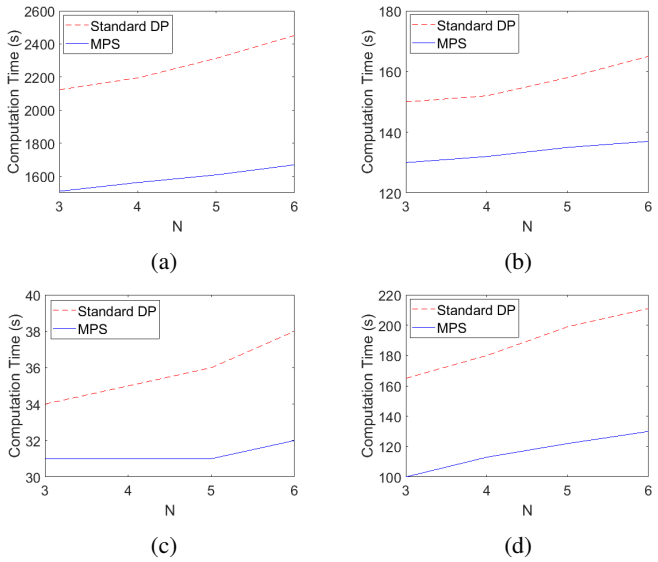


Fig. 14: Comparison of computation time vs $N$ by MPS: (a) UCF-101 dataset. (b) J-HMDB dataset. (c) UCF-Sports dataset. (d) LIRIS-HARL dataset

conducts the search process once instead of several times by the standard DP. This savings is attained at the price of higher storage requirements in each node. Note that the improvement of the search speed on the J-HMDB and the UCF-Sports datasets is not as significant as that on the UCF-101 and LIRIS-HARL datasets, as the former has fewer video frames than the latter. The complexity savings will become more significant when there are more bounding boxes and frames.

Next, we compare the computation time of MPS and the standard DP versus the number of paths, $N$, on all datasets, as shown in Fig. 14, from which we can see that the complexity of both algorithms increases with $N$, as more paths will incur higher complexity. We also find that the proposed MPS is faster than the standard DP for all $N$s. The time parsimony is because MPS effectively reuses the information generated in the forward message passing and then traces back all of the possible action paths in only a single run, instead of restarting DP to find one path after another. Again, the improvement of computation time on the J-HMDB and the UCF-Sports dataset is less significant as the videos in these two datasets are shorter.

## V. Conclusions

This paper developed an effective algorithm for multiple action tube detection in videos. An efficient MPS method is addressed, which fully reuses the accumulated scores and the path information determined in the forward message passing to simultaneously find all multiple paths in the backward path tracing. Additionally, a novel VLR scheme is addressed to rectify the potentially inaccurate bounding boxes, and a new fusion strategy is devised to circumvent the adverse small camera motion effect to render more accurate action detection scores. Both schemes can further enhance the detection accuracy of MPS. Simulations on the UCF-101 and the J-HMDB datasets validate the proposed approach.

## Acknowledgment

We would like to thank Mr. Saha for many useful suggestions and for providing the source code of [4] used in our simulations. We also would like to express our gratitude to the associate editor and the reviewers for many thoughtful comments and suggestions which have enhanced the quality and readability of this paper. The help of Mr. Rizard Renanda Adhi Pramono in running some simulations is also greatly appreciated. This work was supported by National Ministry of Science and Technology, R.O.C. under contracts MOST 106-2221-E-011-140 and MOST 107- 2221-E-011-078-MY2.

## References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of the Neural Information Processing Systems*, 2015, pp. 91–99.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[3] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[4] 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 *Proceedings of the British Machine Vision Conference*, 2016, pp. 58.1–58.13.

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

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

[7] G. Yu and J. 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] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 219–233.

[9] D. Oneata, J. Verbeek, and C. 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] Y. Tian, R. Sukthankar, and M. 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] Z. Shu, K. Yun, and D. Samaras, "Action detection with improved dense trajectories and sliding window," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 541–551.

[12] J. C. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek, "Apt: Action localization proposals from dense trajectories," in *Proceedings of the British Machine Vision Conference*, 2015, pp. 177.1–177.12.

[13] R. Sibson, "Slink: an optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.

[14] P. Mettes, J. C. van Gemert, and C. G. M. Snoek, "Spot on: Action localization from pointly-supervised proposals." in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 437–453.

[15] L. Wang, Y. Qiao, X. Tang, and L. Van Gool, "Actionness estimation using hybrid fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2708–2717.

[16] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2904–2913.

[17] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *Proceedings of the IEEE European Conference on Computer Vision*, 2016, pp. 744–759, part of the results is from https://github.com/pengxj/action-faster-rcnn.

[18] Z. Yang, J. Gao, and R. Nevatia, "Spatio-temporal action detection with cascade proposal and location anticipation," in *Proceedings of the British Machine Vision Conference*, 2017.

[19] E. H. P. Alwando, Y.-T. Chen, and W.-H. Fang, "Multiple path search for action tube detection in videos," in *Proceedings of the IEEE International Conference on Image Processing*, 2017.

[20] C. Ming-Ming, Z. Zhang, W. Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3286–3293.

[21] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2417–2424.

[22] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 222–234, 2014.

[23] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized prim's algorithm," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2536–2543.

[24] J. Carreira and C. Sminchisescu, "Cpmc: Automatic object segmentation using constrained parametric min-cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1312–1328, 2012.

[25] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[26] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 391–405.

[27] H. Wang, A. Klser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3169–3176.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 818–833.

[31] R. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 1440–1448.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository*, vol. abs/1409.1556, 2014.

[33] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the International Conference on Machine Learning*, 2014, pp. 647–655.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *in Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[35] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[36] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv preprint arXiv:1611.10012*, 2016.

[37] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[38] I. Laptev and P. Pérez, "Retrieving actions in movies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[40] A. Klaser, M. Marszaek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 275.1–275.10.

[41] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, and G. Wang, "Real-time obstacle detection and tracking for sense-and-avoid mechanism in UAVs," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 185–197, 2018.

[42] V. Veeriah, N. Zhuang, and G. J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4041–4049.

[43] H. Hu, Z. Wang, J. Y. Lee, Z. Lin, and G. J. Qi, "Temporal domain neural encoder for video representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2192–2199.

[44] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "Videolstm convolves, attends and flows for action recognition," *Computer Vision and Image Understanding*, vol. 166, pp. 41 – 50, 2018.

[45] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6158–6166.

[46] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *in Proceedings of the Advances in Neural Information Processing Systems*, pp. 568–576, 2014.

[47] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4768–4777.

[48] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen, "Temporal context network for activity localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5793–5802.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[50] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of the International Conference on Learning Representations*, 2014.

[51] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[52] K.-W. Cheng, Y.-T. Chen, and W.-H. 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.

[53] S. Gidaris and N. Komodakis, "Locnet: Improving localization accuracy for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 789–798.

[54] D. Comaniciu and P. 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.

[55] 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, 2007.

[56] E. K. Chong and S. H. Zak, *An Introduction to Optimization*. John Wiley & Sons, 2013.

[57] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CRCV-TR-12-01*, 2012.

[58] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the IEEE Conference on Computer Vision*, 2013, pp. 3192–3199.

[59] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[60] C. Wolf, E. Lombardi, J. Mille, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandréa, C.-E. Bichot *et al.*, "Evaluation of video activity localizations integrating quality and quantity measurements," *Computer Vision and Image Understanding*, vol. 127, pp. 14–30, 2014.

[61] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proceedings of the European Conference on Computer Vision*, 2004, pp. 25–36.

[62] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[63] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1134–1142.

[64] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proceedings of the IEEE*

*International Conference on Computer Vision*, 2017, pp. 5822–5831.