Fast and Accurate Action Detection in Videos with Motion-Centric Attention Model

Jinzhuo Wang, Wenmin Wang*, *Member, IEEE* Wen Gao, *Fellow, IEEE*

***Abstract*—A key factor that makes action detection in videos differing from general video classification is human-guided clues especially motion signals. Since not all pixels in a video are informative for action recognition, the irrelevant and redundant parts can bring much noise and extra burden for both feature extraction and classifier training. This encourages researchers to seek the design of attentive model that can dynamically focus computations on the key spatiotemporal volumes. In this paper, we propose a motion-centric attention model for action detection in videos which imitates the human perception of saccade and fixation procedure when detecting actions in a video. Specifically, we first present a strategy to generate motion-centric locations based on the density peak of motion signals, providing reliable candidates around which actions have high possibilities to occur. Then we introduce an attention model which conducts saccade and fixation procedures on these candidates to observe local spatiotemporal visual information, preserves internal comprehension, and produces action proposals on temporal bounds. Afterwards, a classifier with several variants is prepared to classify the action proposals and decide which one to fixate and generate the final predictions. We show how to efficiently train our model to produce fast and accurate action detection, by only scanning a small fraction of locations in a video. Extensive experiments on three challenging datasets show promising results in both accuracy and speed.**

***Index Terms*—Action detection, motion-centric, attention model, recurrent neural network**

I. Introduction

ACTION detection in videos is a challenging problem, and has drawn increasing interests in computer vision and multimedia community due to its potential applications in video surveillance, human computer interaction, video content analysis, *etc*. It is required to determine both the semantic label of an action along with when it starts and ends in a video. Current algorithms typically employ visual information of all frames for action detection (Fig. 1) [1] [2]. In both traditional bag-of-features methods and deep learning models, each frame is processed to extract frame-level features and fed to classifiers exhaustively through the entire video to produce action predictions [3] [4] [5]. The runner up submission in ActivityNet Detection Challenge 2016 also needs to score every frame to assist localization and classification [6]. These approaches are not efficient since the computational complexity inevitably depends on the video size. Besides, the irrelevant frames are a heavy burden for both feature extraction and classifier training. A natural way to escape this problem is to select several key frames and collect features around key frames to represent the entire video [7] [8]. However, this may neglect motion clues in different temporal scales and long-range structures [9] [10].

Recently, human-inspired techniques have achieved remarkable success in a wide range of problems [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22]. We also seek for efficient solutions for action detection from the way of human perception. Previous works [23] [24] suggest that high-acuity vision is restricted to a small foveal region surrounding the current fixation point, with acuity dropping off precipitously from that focal point. In image recognition, the visual system handles this constraint by rapidly reorienting the eyes an average of three times each second via saccadic eye movements [25]. Construction of a complete visual representation would therefore seem to require the storage of a high-resolution image across saccade, with images from consecutive fixations overlapped or spatially aligned to form the composite image [26]. This is consistent to the behavior of eye-movement when searching actions in videos [27]: In most cases, human would not process the entire scene at once. Instead they often selectively focus on some portions to acquire representative information, form an internal presentation by integrating what they have saw, and gradually fixate the region of interest. Utilizing such attention mechanism [28], some efforts have demonstrated effectiveness in image recognition [29], natural language processing [30] and speech recognition [31]. We extend in spatiotemporal domain and study its potential for action detection in videos.

In this paper, we propose a motion-centric attention model that can automatically localize and recognize actions in long, untrimmed videos by scanning only a few fragments for a video. Our model mainly consists of three parts, i.e. a motion-centric location generator, an attention network and a classifier. The first part is generating motion-centric locations in an unsupervised manner using the density of local descriptors. These generated candidates serve as a prior knowledge to prevent the following attention network from distraction of irregular visual information, which we show can contribute significant improvement on both accuracy and speed. The second part is an attention network which aims to generate high recall temporal proposals. Our intuition is that observing frames at a few positions in a video can gradually narrow down the extent where an action might occur, as shown in Fig. 1. Moreover, visual representation extracted only from the generated proposal is sufficient for a classifier to further distinguish the foreground and background. To this end, our attention network is implemented around a recurrent neural network (RNN), which takes a location and the corresponding

Jinzhuo Wang, Wenmin Wang and Wen Gao are with the School of Electronic and Computer Engineering, Peking University, China. Corresponding author is Wenmin Wang. Email address: wangwm@ece.pku.edu.cn

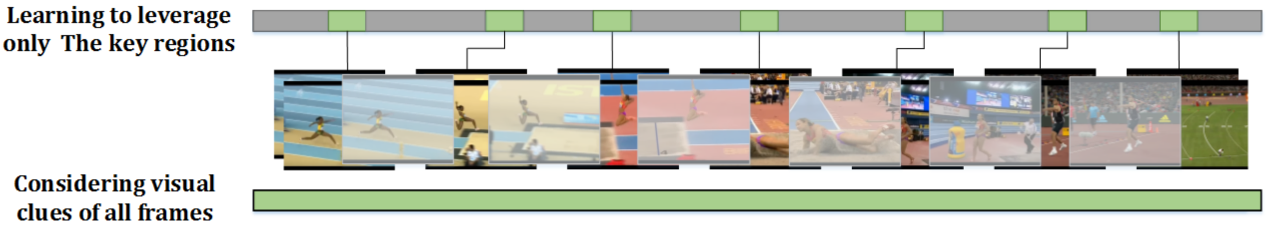

Learning to leverage only The key regions
Considering visual clues of all frames

Fig. 1. Comparison with most existing action detection strategies. The proposed method (up) only observes a small fraction of continuous frames (green part) while most recent approaches (down) rely on visual clues of the entire video that need to compute features of all frames.

video fragment as input and outputs a possible action bound at each time step. The third part is an off-line classifier. After the temporal proposals are generated, the classifier encodes a few frame-level features from each proposal and outputs scores for all classes. We experiment several classifiers with different training strategies and show clear difference. The proposed method can successfully escape the distraction of irrelevant visual signals, and consequently obtain a fast and accurate prediction of actions. We show the efficiency of our system in both accuracy and speed on three challenging large-scale benchmarks. Compared with most recent approaches for action detection, our method needs less runtime and produces more reliable predictions.

The main contributions of this paper are summarized as follows: First, we propose a motion-centric attention model for action detection. Relevant works often integrate location transfer and visual recognition in the attention model, we show such setting may led to ambiguous semantic understanding since localization requires global information while classification requires visual clues of specific actions. Besides, we introduce several good practices such as classifier design and training manners using attention model for action detection, and demonstrate their effectiveness over pervious methods. Second, we introduce a technique to generate location candidates around which an action has high possibilities to occur. Compared with existing proposal extraction methods that need complex computation referring to ground truth, the proposed method performs in an unsupervised manner and can produce high quality candidates. This can help attention model jump under a reasonable subset instead of stochastic space in previous attention-based methods. Third, we conduct extensive experiments on three large-scale benchmarks and report competitive results in terms of both accuracy and speed.

The rest of this paper is structured as follows. Section II reviews relevant works on action detection especially using motion mining and visual attention, followed by our proposed model and detailed implementations in Section III. Then, we provide experimental results and comprehensive analysis in Section IV. Finally, we conclude our paper in Section V.

II. Related Work

There is a long history of work in video analysis and action detection [32] [33] [34] [35] [36] [37] [38]. In the following we first review related works from the perspectives of hand-crafted features and deep learning models. Then we review attention-based models which is similar to our method, especially their use in action detection. For more comprehensive studies, we refer to surveys [39] [40] [41].

A. Hand-crafted feature for action detection

Early action detection methods rely heavily on hand-crafted features. In particular, [42] first extended 2D Harris corner detector to obtain representative tubes in 3D space. Since then many 2D local descriptors are extended to 3D version for video understanding such as HOG3D [43] and 3D-SIFT [44]. A comprehensive evaluation in [45] compared different STIP detectors and descriptors. The authors drew conclusions that the performance of STIPs is dataset dependent. Besides, many attempts have been made to explore relevant relationships of STIPs for action recognition, which usually pursue higher order statistics of the already extracted STIPs, such as pairs [46], groups [47], point clouds [48], alignment [49] and clusters [50]. Recently, [51] made use of point trajectories to extract and align 3D volumes, and resorted to more rich low level descriptors for constructing effective video representations, including HOGHOF and MBH. An improved version of dense trajectory is updated in [1] to estimate camera motion, and obtained state-of-the-art results on a variety of benchmarks.

Although local hand-crafted features yield promising results, one limitation is that they lack semantics and discriminative capacity. To overcome this issue, several mid-level and high-level video representations have been proposed such as Action Bank [52], Dynamic-Poselets [53], Actons [54], Tubelets [55]. They usually resorted to some heuristic mining methods to select discriminative visual elements as feature units. But these methods still need to compute visual signals of the entire video, while our system can escape these limitations by only scanning a few number of specific locations.

B. Deep learning for action detection

In contrast to the hand-crafted features, there is a growing trend of learning features directly from raw data using deep learning techniques, which has achieved great success in image-based tasks [13] [56] [57]. A number of attempts have developed deep architectures especially convolutional neural network (CNN) for video action detection [2] [58] [59] [60] [61] [62] [63] [64] [65]. In particular, [59] extended 2D ConvNet to video domain by stacking static frames for action recognition on relatively small datasets, and recently [58] tested similar deep networks on a large dataset (Sports-1M). However, these deep models achieved lower performance compared with shallow hand-crafted representation [1], which might be ascribed to the following reasons. Firstly, available action datasets are relatively small for deep learning. Secondly, learning complex motion patterns is more challenging. Most CNN-based approaches rely on the neural networks to perform the final class label prediction, normally using a softmax layer [2] [58] or a linear layer [59]. Instead of direct prediction by deep neural networks, [66] conducted action recognition using support vector machines (SVMs) with features extracted from off-the-shelf CNN models. Their impressive results in the THUMOS action recognition challenge [67] indicate that CNN features are very powerful. Very recently, [60] proposed C3D spatiotemporal features learnt from carefully designed deep convolutional networks and demonstrated competitive performance with dense trajectories.

In addition, a few works apply the CNN representations with RNN models to capture temporal information in videos and perform classification within the same network. [68] [69] [70] leveraged RNN model with LSTM units for action recognition and [71] proposed to translate videos directly to sentences with the LSTM model by transferring knowledge from image description tasks. Combining RNNs with CNNs for video understanding also shares the same motivation with the temporal pathway in the popular two-stream framework [2]. Our method also uses RNN structure but within an attention model.

C. Attention model for action detection

Recently visual attention model is extremely popular, which aims to capture the property of human perception mechanism by selectively observing and consequently identifying the interesting regions in a scene. A recent survey reviews RNN-based attention models and their applications to computer vision tasks [72]. Also, there are many attention-based models proposed for video analysis [73] [74] [75] and action recognition [76] [77] [78] [79] [80] [81] [82]. Our model makes non-trivial efforts and differs from them in the following aspects: First, existing attention models such as [29] [83] [75] [77] [81] tends to train visual analyzer and attention policy in an end-to-end manner, sharing information from hidden states of the recurrent neural network. However, it may lead the model to ambiguous semantic understanding since localization requires global information while classification requires visual clues of specific actions. Based on this insight, our model uses serial design, and employs an attention model to fetch appreciate action proposals used for more precise classification with a well-trained classifier. We provide several types of classifiers and training manners, demonstrating the advantages over previous strategies. Second, at each training step, we choose to observe a short fragment of several continuous frames rather than just a single frame [75] [80] [77]. This choice aims at making use of both spatial and temporal visual information, which we show can lead to more precise detection. Third, instead of stochastic attention over the entire searching space which is used in most attention models [29] [83] [75] [74] [78], we introduce a novel clustering-based method to generate sparse yet reliable location candidates using the density peak of motion signals. These candidates concentrate on motion-centric regions, providing reliable initialization and searching space for our attention model.

III. Model

In this section, we first give an overview of our system in Section III-A, where the execution flow is offered along with component relations. Then, we present the key components including the model structure and training manner. More specifically, Section III-B introduces a technique to generate motion-centric location candidates based on the density of local descriptors, which are prepared for the attention model. Then, we build our attention network used to produce action proposals in Section III-C. Afterwards, in Section III-D, we introduce an off-line classifier used to analyze the generated action proposals and produce the final prediction.

A. Overview

Our model mainly consists of three parts: a motion-centric location generator, an attention network and a classifier, as shown in Fig. 2. In particular, given an input video, we first generate a few motion-centric locations using density information of local spatiotemporal descriptors in an unsupervised way. These candidates have high possibility to detect actions since most action instances occur around locations containing dense motion signals. More importantly, they serve as a prior knowledge to prevent our attention network from distraction of irregular regions, which we show can contribute significant improvement on both accuracy and speed. The attention network is designed to produce high recall action proposal. It has four subnetworks responsible for observing location candidates, preserving internal comprehension, generating temporal bounds and determining next location, separately. The third part is an off-line classifier. After the action proposals are generated, the classifier encodes a few frame-level features from each proposal and outputs scores for all classes to determine the final prediction of the highest confidence. We design several classifiers with different training strategies and show clear advantages over other choices in existing attention-based models.

B. Motion-centric location generator

Videos are well known containing much redundant information among consecutive frames. Such issue is often dealt with hand-crafted solutions such as frame sampling [10] and motion extraction [84] [85]. However, these methods may not be suitable in our case since our attention model is expected to receive locations around which actions have high confidence to occur. Here we introduce a technique based on motion density to obtain location candidates, which we show have high probabilities to contain key actions and also be suitable for the following stages. This approach can also be regarded as a general prior step for other video-specific tasks such as saliency detection.

We first extract a set of motion signals of the input video which is optical-flow-based feature in spatiotemporal domain. In practice, we use dense trajectory [51] (comparison of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2887061, IEEE Transactions on Circuits and Systems for Video Technology

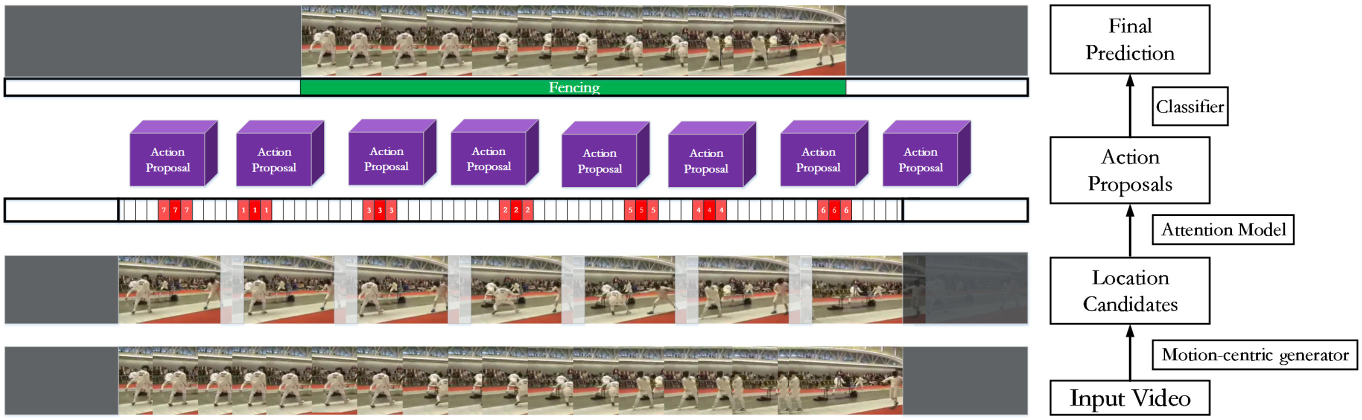


Fig. 2. Illustration of the overall system. From bottom to up is the execution procedure. Given an input video, we first generate some motion-centric location candidates using density information of local spatiotemporal descriptors. We then extract local spatiotemporal features around these candidates prepared for our attention model. The model sequentially processes these observations and produces the action proposals which are sent to well-trained classifier to generate final prediction for the task of action detection. The detailed structure of attention model can be seen in Fig. 4.

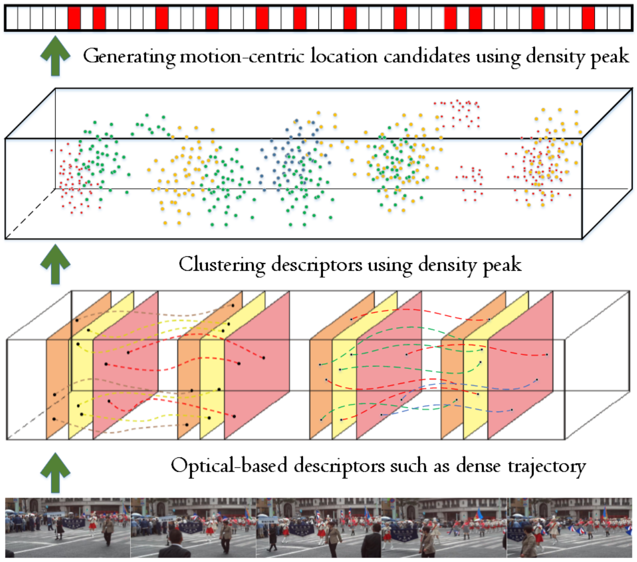

Generating motion-centric location candidates using density peak
Clustering descriptors using density peak
Optical-based descriptors such as dense trajectory

Fig. 3. Location candidates generated with density peak of motion signals. From bottom to up: Given a video, we first extract local descriptors (performance comparison can be seen in Table I), which are then clustered using its density peaks. Afterwards, we obtain the location candidates corresponding to the clustering center, which is ready for our attention model to jump upon to produce action proposals.

other local features can be seen in Table I). Next we cluster these motion signals to obtain a distribution over the entire spatiotemporal space and utilize the density peaks to generate location candidates. Since motion signals are of high number and dimension, it is difficult to define its category number in advance. We introduce a clustering approach which requires its basis only in the distance between data points. We are going to employ density peaks to pursue a reasonable clustering and in turn, leverage these peaks for candidate generation.

The algorithm has its basis in the assumptions that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density [86]. In practice we find Euclidean distance is the best choice. For each feature i, we compute two quantities: its local density r_i and its distance d_i from features of higher density. Both these quantities depend only on the distances d_{ij} between two features i and j, which are assumed to satisfy the triangular inequality. We define the local density r_i of feature i as

$$r_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a cutoff distance. Basically, r_i is equal to the number of points that are closer than d_c to point i. This procedure is sensitive only to the relative magnitude of r_i in different features, which means the results of the analysis are robust with respect to the choice of d_c for large sets. In addition, ψ_i is measured by computing the minimum distance between feature i and any other point with higher density

$$\psi_i = \min_{j:r_j>r_i} (d_{ij}) \tag{2}$$

For the feature with the highest density, we set $\psi_i = \max_j(d_{ij})$ over the entire set of descriptors. Note that ψ_i is much larger than the typical nearest neighbor distance only for points that are local or global maxima in the density. Thus, cluster centers are recognized as features for which the value of ψ_i is anomalously large. Finally we select the density peaks as our location candidates, prepared for the following stages. A typical processing procedure can be seen in Fig. 3.

C. Attention model

The goal of our attention model is to take a set of location candidates which are generated in Section III-B and output any detected action instances. As discussed in Section I, we expect it can gradually attend the region of interest by scanning only a few fragments in a video. This procedure requires to preserve an internal comprehension of its continuous observations in discrete steps. Thus we choose to formulate our attention network as an RNN h_t that interacts with observed location candidates over time t, as shown in Fig. 4.

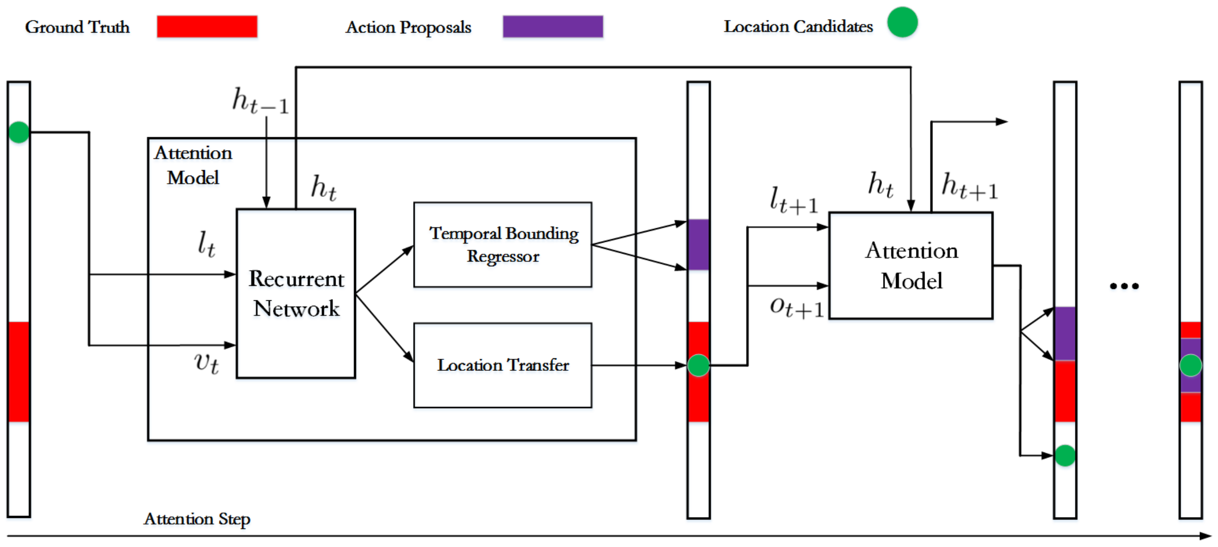


Fig. 4. Illustration of our attention model. The center is an recurrent neural network. At each time step, it observes current location candidate, and receives three parts: current location l_t, visual clues v_t and its last state h_{t-1}. On the other hand, it outputs two parts: an action proposal (s_t, e_t) as well as the next location l_{t+1}. The figure shows a typical procedure for several attention steps from left to right.

As can be seen in Fig. 4, at each time step t, the network receives the current candidate location l_t and the observation feature vector v_t around the location candidate in a local spatiotemporal region. In practice, we use the fc7 feature extracted from a pre-trained C3D network [60]. Both of them serve as a part of the input of RNN's hidden state h_t. In addition, the other input of h_t is its previous state h_{t-1}, used to accumulate and integrates what it has observed and the temporal and semantic hypotheses about action instances. The model is expected to gradually exclude locations we are not interested in, and narrow down where an activity might occur.

The outputs of our attention model are divided into two parts. The first one is a temporal prediction using a temporal bound regressor. We design it with fully connected layers. The regressor propagates h_t and directly predicts a tuple $p_t(s_t, e_t)$ which is normalized into $[0,1]^2$ as a temporal proposal of an action instance, where s_t and e_t are the start and end locations, respectively. The second is the next location candidate to observe which is governed by an attention policy. Since we would not examine every possible candidate, such policy needs to be learned with reinforcement learning, namely soft attention mechanism as discussed in Section II-C. In the following we describe the training manner of the two parts.

Training. In recent attention-based video analysis works [74] [75] [76] [88] [77] [78] [79] [80] [81], the generation and evaluation of proposals are conducted simultaneously. Although the internal comprehension of RNN can be utilized for multi-task for an attention-based model, however, in our perspective, extra information from previous RNN steps is not necessary to classify the current proposal, as good classifier can be pre-trained from large scale database, in a off-line manner. Therefore, to generate high quality proposals and classify them more accurately, we train the attention model and classifier separately, where the latter is described in Section III-D.

As addressed above, the attention network produces an action bound prediction $p_t(s_t, e_t)$ and a location l_t at each time step. When training the temporal bounding regressor, a ground truth action segment needs to be selected to refine the prediction at each time step. Given a set of ground truth $G = \{g_i\}$ for a video, for each p_t, if there exists at least one g_i overlapping with p_t, the one with a maximum overlap is chosen as the ground truth $g_c(s_c, e_c)$. Otherwise, the one with a minimum distance with p_t should be selected. Besides, for each pair of p_t and g_i at time step t, a matching indicator m_{ti} is set 1 if g_i has the largest overlap with g_c, while $m_{ti} = 0$ for other situations. The overlap and distance function are defined as

$$\text{overlap}(p_t, g_i) = \frac{\min(e_t, e_i) - \max(s_t, s_i)}{\max(e_t, e_i) - \min(s_t, s_i)} \tag{3}$$

$$\text{dist}(p_t, g_i) = \min(|s_t - e_i|, |e_t - s_i|) \tag{4}$$

where dist measures the distance between two video clips [78]. Once g_c is chosen, the loss function for the temporal bounding regressor can be formulated using a smooth $\mathcal{L}_1$ loss

$$\mathcal{L}_{loc}(p_t, g_c) = \text{smooth}_{\mathcal{L}_1}(s_t - s_c) + \text{smooth}_{\mathcal{L}_1}(e_t - e_c) \tag{5}$$

where $\text{smooth}_{\mathcal{L}_1}$ is a widely used loss function less sensitive to outliers [89]

$$\text{smooth}_{\mathcal{L}_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{6}$$

Since attention policy is in a non-differentiable setting (we would not examine every possible candidate) and conventional back-propagation is not adequate here, we turn to reinforcement learning which evaluates each decision by a reward, and learns to find the optimal sequence of observation location with highest cumulate reward. At each single step, a scalar reward r_t is evaluated. If $\text{overlap}(p_t, g_c)$ is larger than a threshold (which is set 0.5 in our work), 1 will be assigned to r_t, otherwise r_t is set -0.1. The cumulative reward $\mathcal{R}_t$ after

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2887061, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 5. Sample frames from MPII-Cooking (top), THUMOS'14 (middle) and ActivityNet (bottom) datasets. The first two datasets contain more simple, background-fixed actions while ActivityNet contains more complex, multi-person videos.

t steps is formulated as $\mathcal{R}_t = \sum_{j=1,\cdots,t} r_j$. The goal of the training procedure is to minimize the difference between the predictions and the ground truth as well as maximize the final cumulative reward $\mathcal{R}_T$. We define the total loss function over the prediction set $\mathcal{L}(P)$ as following

$$\mathcal{L}(P) = \sum_t \sum_i [m_{ti} = 1]\mathcal{L}_{loc}(p_t, g_i) - \lambda \mathcal{R}_T \tag{7}$$

where the hyper-parameter λ is set 1 in all experiments. We use reinforcement learning with REINFORCE algorithm to optimize this loss function.

D. Classifier

Once we obtain the temporal bounds of action proposals, we are going to classify them using our off-line well-trained classifiers. Some recent attention-based approaches such as [75] [74] [77] directly use features from hidden states for visual prediction. However, what the model has observed may be action-unrelated or contain different actions so these features are farraginous and inappropriate for classifying specific actions. Instead, a softmax classifier is used in our work to classify temporal proposals. The classifier simply consists of two fully connected layers, a ReLU layer and a softmax layer. The classifier receives C3D features from a short video fragment around the action proposals and outputs a probability distribution for all action classes. We experiment several training strategies on our classifier to explore good practices for better performance of action detection.

In practice, to train a reliable classifier, we first trim the videos of training set into foregrounds and backgrounds, which are used for positive and negative samples respectively. Next we design three kinds of training strategies.

- "Normal": For each positive or negative trimmed video, we randomly sample n local spatiotemporal features and concatenate them into a feature vector. The classifier then propagates this feature vector and outputs scores for K classes along with the background. Typically, n is set 5 throughout our experiment.
- "Left+Right": Based on "Normal", on the left and right to each trimmed video, we randomly generate 2 more video segments as training samples. These two segments have a high overlap, which is set 0.8, with the trimmed video, thus they share the same label. Since the prediction of attention model tend to contain both foreground and background information, when trained with these generated segments including similar information, classifier may perform a more precise classification performance.
- "Only-Positive": Ignoring the negative samples, we randomly sample one local spatiotemporal feature from each positive trimmed video to train a K-class classifier. At test-time, the features extracted from the middle location of the proposal is used to perform classification.

In addition to the softmax classifier, we also train an one-vs-all linear SVM for each action class. When training a class-specific SVM classifier, we utilize all trimmed videos of other categories as negative samples, including foregrounds and backgrounds. Experiments shows the results and comparison of different settings in Section IV-C (see Table III and II for details).

IV. Experiments

A. Datasets and evaluation protocols

We test our method on three challenging datasets, i.e. MPII-Cooking, THUMOS'14 and ActivityNet. Sample examples of video frames are illustrated in Fig. 5.

The **MPII-Cooking** [90] is a large fine-grained cooking activities dataset. It contains 44 videos with a total length of more than 8 hours of 12 participants performing 65 different cooking activities. It consists of a total of $5,609$ annotations spread over the 65 activity categories, including a background class for the action detection task. Following the standard protocol in [90], we have 7 splits after performing leave-one-person-out cross-validation. Each split uses 11 subjects for training, leaving one for validation.

The **THUMOS'14** [91] dataset is considered to be one of the most challenging datasets for action detection, which is dedicated to localizing action instances in long untrimmed videos. The trimmed videos used for training are $2,755$ videos of these 20 actions in UCF101 dataset [92]. The validation set contains $1,010$ untrimmed videos with temporal annotations of $3,007$ instances in total. The test set contains $3,358$ action instances from 1574 untrimmed videos, whereas only 213 of them contain action instances of interest. We exclude the remaining $1,361$ background videos in the test set.

The **ActivityNet** [93] dataset comprises 28K videos of 203 activity categories collected from YouTube. It consists of 68.8 hours of temporal annotations in 849 hours of untrimmed, unconstrained video. There are 1.41 action instances per

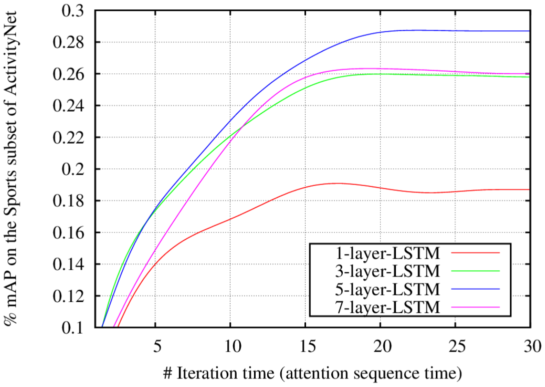


Fig. 6. Performance comparison on the "Playing sport" subset of ActivitNet dataset using different RNN structures as our attention model. Note that all the structures has 256 of unit for each layer.

TABLE I
PERFORMANCE (%) IMPACT OF DIFFERENT LOCAL SPATIOTEMPORAL DESCRIPTORS FOR CANDIDATE GENERATION. ON "PLAYING SPORTS" SUBSET OF ACTIVITYNET DATASET.

Hand-crafted	mAP	CNN-based	mAP
iDT [1]	**25.2**	C3D [60]	13.4
HOG3D [43]	19.7	Deep Two-Stream CNNs [10]	21.4
3D-SIFT [44]	18.5	Hidden Two-Stream CNNs [95]	22.5

video and 193 instances per class. More importantly, many activities are relatively long and complex, and the viewpoint and foreground objects may change significantly within the same activity. The authors of ActivityNet use one fourth of the dataset as a validation set, but have not released the test set used in their paper. In our experiments, we use the validation set as our test set following [94].

Evaluation metric. We follow the conventional metrics used in temporal action detection task to regard it as a retrieval problem, and evaluate mean average precision (mAP). A prediction P is marked as correct only when it has the correct category prediction, and has intersection over union (IOU) with ground truth G larger than the IOU threshold α

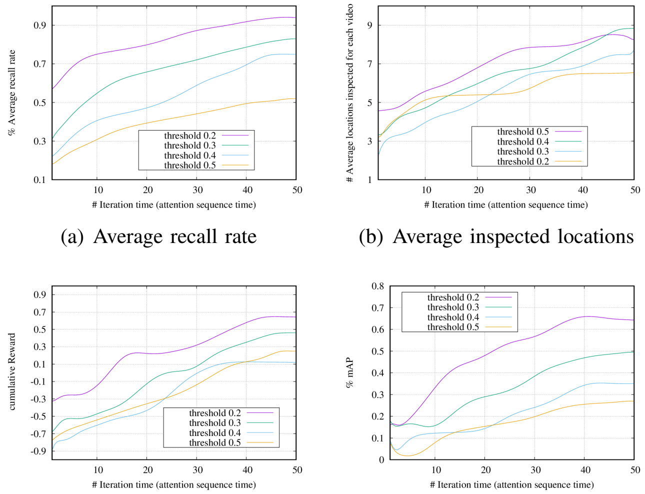
$$\alpha(G,P) = \frac{1}{|\phi|}\sum_{f\in\phi}\frac{G_f \cap P_f}{G_f \cup P_f} \tag{8}$$

where ϕ is the test set. Following most relevant works, we conduct experiments with various of α from 0.1 to 0.5.

B. Implementation details

This part presents the implementation details of our model. Our model is implemented using Torch7 framework [96]. For the stage of motion-centric location candidate generation, we test three kinds of optical-flow-based descriptors as well as three kinds of CNN-based features.

- iDT [1]: The size of the each volume in a video is $n \times n$ pixels and L frames. The volume is subdivided into a spatiotemporal grid of size $n_\sigma \times n_\sigma \times n_\tau$. We use the default parameters $n = 32, L = 15, n_\sigma = 2, n_\tau = 3$. Trajectory is constructed with the sampled points in the volume using a dense optical flow field. iDT is represented with HOGHOF [97] along the dense trajectories as recommended in [98].
- HOG3D [43]: Each video clip is divided into $n_x \times n_y \times n_t$ cells. The corresponding descriptor concatenates gradient histograms of all cells and is then normalized. We use the executable from the authors website[1] and apply their recommended parametric settings for all feature detectors: descriptor size $\Delta_x(\sigma) = \Delta_y(\sigma) = 8\sigma), \Delta_t(\tau) = 6\tau$, number of spatial and temporal cells $n_x = n_y = 4, n_t = 3$, and icosahedron as polyhedron type for quantizing orientations.
- 3D-SIFT [44]: We use Harris detector to extract interest points with the same volume size in video as iDT. The sizes of cube and sub-cube for each point are $12 \times 12 \times 12$ and $2 \times 2 \times 2$. The 3D-SIFT is formed by combining all the unit cube histograms [44].



(a) Average recall rate

(b) Average inspected locations

(c) Cumulative reward

(d) mAP

Fig. 7. Curves of different factors during the procedure of training our attention model on the "Playing sport" subset of ActivitNet dataset. All measurements gradually become stable as iteration increases.

We also cover three CNN-based deep features: C3D [60], Deep Two-Stream CNNs [10] and Hidden Two-Stream CNNs [95]. We choose the last feature maps before fully connected layers. For each of the above six types, we collect 3,000 descriptors for each video, resize them to 100-dimension and employ the method in Section III-B to obtain location candidates. The rest of clustering implementation for clustering follows the instructions of [86].

As for our attention model, we use the RNN-based model from [29] as the basic architecture, adapting it to action detection task in video. Specifically, we test four kinds of LSTM networks (1, 3, 5, 7 layer) with 256 hidden units each. The attention sequence number is experimented to pursue a good number. The model receives C3D features [60] as visual clues of local spatiotemporal region around the motion-centric location candidates. Each C3D feature is generated at 2 frames per second with temporal resolution of 16 frames, which is

[1] http://lear.inrialpes.fr/software

TABLE II
PERFORMANCE (%) COMPARISON USING DIFFERENT CLASSIFIERS ON “PLAYING SPORTS” SUBSET OF ACTIVITYNET DATASET WITH IOU THRESHOLD $\alpha = 0.5$.

	“Normal”	“L+R”	“Only-Pos”	SVM	“Joint”
High jump	13.26	13.59	17.26	18.67	7.44
Long jump	4.05	5.21	6.68	1.86	0.46
Cricket	19.20	19.10	16.07	13.63	9.40
Discus throw	1.22	6.65	5.98	1.11	8.99
Rollerblading	19.46	26.98	23.27	27.15	7.97
Powerbomb	33.51	48.79	45.59	61.98	31.17
Javelin throw	3.03	6.71	6.45	7.08	5.04
Longboarding	28.44	25.80	21.69	31.03	15.04
Hurling	29.90	21.94	24.25	30.54	22.85
Shot put	2.06	2.40	0.44	1.38	1.32
Paintball	14.22	16.48	20.29	22.18	12.26
Bungee jump	4.83	8.74	4.27	6.08	0.34
Triple jump	7.50	4.04	4.70	7.70	2.08
Pole vault	11.73	7.23	6.46	4.69	2.43
Powerbocking	37.26	36.26	43.39	43.09	31.53
Croquet	23.72	23.76	27.24	25.64	15.11
Hammer throw	10.13	16.44	15.29	5.33	4.00
Skateboarding	21.61	17.00	18.26	20.59	2.37
Dodgeball	41.26	49.71	55.63	63.72	72.06
Doing moto	48.21	46.47	54.73	60.53	45.03
Starting camp	45.44	44.48	49.50	62.50	38.20
Archery	13.28	12.44	21.71	15.16	9.94
Camel ride	78.96	69.38	70.99	84.25	51.74
Playing ball	23.30	37.15	34.94	39.58	22.29
Baton twirling	72.30	71.27	71.15	78.19	60.62
Curling	14.50	7.10	11.94	14.37	9.57
mAP	23.94	24.81	26.08	**28.77**	18.82

TABLE III
PERFORMANCE (%) COMPARISON USING DIFFERENT TRAINING MANNERS ON “PLAYING SPORTS” SUBSET OF ACTIVITYNET DATASET WITH IOU THRESHOLD $\alpha = 0.5$.

	U-single	U-ova	S-single	S-ova
High jump	7.44	7.55	19.62	13.99
Long jump	0.47	6.38	1.86	6.37
Cricket	4.43	12.15	12.67	15.26
Discus throw	4.53	11.23	1.11	10.08
Rollerblading	5.33	10.87	24.74	22.62
Powerbomb	25.19	48.81	61.98	61.05
Javelin throw	5.04	2.07	7.08	14.64
Longboarding	18.22	32.52	37.00	38.56
Hurling	15.19	28.11	30.54	24.16
Shot put	1.32	0.29	1.38	0.57
Paintball	11.30	18.75	22.18	29.03
Bungee jump	0.34	5.09	6.08	15.53
Triple jump	1.19	11.58	7.70	16.89
Pole vault	1.39	12.72	5.06	6.68
Powerbocking	31.53	47.40	43.09	44.48
Croquet	15.11	15.64	25.64	28.96
Hammer throw	0.50	9.75	5.33	14.21
Skateboarding	1.89	12.91	20.59	20.45
Dodgeball	47.50	59.76	63.72	69.01
Doing moto	40.69	51.82	59.21	61.47
Starting camp	32.00	36.23	62.50	62.38
Archery	2.79	15.27	14.38	9.03
Camel ride	45.54	36.01	83.61	87.91
Playing ball	22.29	27.19	39.58	32.67
Baton twirling	53.58	77.03	77.64	75.19
Curling	4.94	0.58	14.37	16.08
mAP	15.37	22.99	28.79	**30.43**

further reduced into 500 dimension using PCA. The temporal bound regressor is implemented with two fully connected layers $(256 \rightarrow 100 \rightarrow 100 \rightarrow 2)$. Parameters of all networks are initialized using uniform distribution between -0.1 and 0.1. During training phase, we use 20 videos as a mini-batch in every training iteration and update parameters with batch gradient descent approach. The learning rate is initially set as 0.01, and decays by 0.0001 per 500 iterations. The momentum is set 0.9. Since detections from different steps may be duplicate, we apply the non-maximum suppression [99] to eliminate redundancy at testing time.

C. Model Investigation

We first evaluate two experimental deployments and determine the best common settings for our model. Then we study the impact of different configurations of attention model and investigate the optimal choice. These experiments are conducted on the “Playing sports” subset of ActivityNet dataset. Afterwards, we report the performance of our best model and compare with other competitive approaches on all three datasets in both accuracy and speed.

Evaluation of motion-centric generator. Based on clustering implementation of [86], We find the most important factor for performance is the choice of optical-flow-based descriptors. We test four types of popular local spatiotemporal descriptors and list the performance comparison in Table I. Note that it uses our attention model with 1-layer LSTM which iterates 15 times for each video. As Table I shows, iDT performs better than the other three choices. Interestingly, hand-crafted descriptors (HOG3D and 3D-SIFT) always achieve better than deep learned features in our method. This suggests that hand-crafted features are more suitable for preprocessing tasks.

Evaluation of attention model structure. This part investigates the optimal RNN-based architecture for our attention network, including the number of layer and the iteration time of attention sequence for each video. As Fig. 6 demonstrates, the performance typically becomes weaker after around 20 locations for each video. Besides, a 5-layer LSTM structure is shown to be the best choice for our attention model. We can also conclude that further increasing the iteration time contributes no improvement, which may be due to the limited training data.

Analysis of attention procedure. We analysis the attention procedure by watching a series of curves of selected key factors with training steps in Fig. 7. In particular, Fig. 7(a) shows the average recall rate with increasing iterations. Fig. 7(b) depicts the average proposals which is processed during the iteration. We can conclude that our model is able to produce fairly short and effective search patterns with less than 10 locations inspected under various conditions. Fig. 7(c) is the curve of cumulative reward of our reinforcement learning method. Fig. 7(a) is the mAP changes with increasing training iterations which shows our model can learn efficient detection policy and finally converge to a stable result. In this figure, we see that all these factors gradually becomes stable during the training procedure.

Evaluation of classifier. As discussed in Section III-D, we have designed 4 types of classifier totally, including 3 types of implementations of softmax classifier along with additional SVM for each class. To evaluate the advantages of our separate

TABLE IV
PER-CLASS PERFORMANCE (%) ON THUMOS-14, AT IOU OF $\alpha = 0.5$ ON THUMOS'14 DATASET.

Class	INRIA [100]	CUHK [3]	[75]	Ours	Class	INRIA [100]	CUHK [3]	[75]	Ours
Baseball Pitch	8.6	16.4	14.6	**19.6**	Hamm. jump	34.7	29.3	28.9	**43.2**
Basket. Dunk	1.0	0.1	**6.3**	5.8	High Jump	17.6	9.7	33.3	**37.1**
Billiards	2.6	2.2	9.4	**12.6**	Javelin Throw	**22.0**	6.2	20.4	6.5
Clean and Jerk	13.3	9.4	**42.8**	35.1	Long Jump	**47.6**	20.0	39.0	25.8
Cliff Diving	17.7	6.3	15.6	**24.7**	Pole Vault	19.6	**17.6**	16.3	13.4
Cricket Bowl.	9.5	0.1	10.8	**26.9**	Shotput	11.9	2.0	16.6	**19.7**
Cricket Shot	2.6	0.4	3.5	**7.0**	Soccer Penalty	8.7	3.6	8.3	**25.4**
Diving	4.6	1.0	**10.8**	3.0	Tennis Swing	3.0	3.5	5.6	**8.3**
Frisbee Catch	1.2	0.2	10.4	**20.5**	Throw Discus	36.2	18.2	29.5	**32.8**
Golf Swing	22.6	23.2	13.8	**31.3**	Volley. Spike	1.4	2.3	5.2	**13.6**
mAP	14.4	8.2	17.1	**21.3**					

TABLE V
ACTION DETECTION RESULTS (%) ON MPII COOKING DATASET.

Method	0.1	0.2	0.3	0.4	0.5
Sliding Window	22.2	19.7	15.8	12.6	7.9
Gemert *et al.* [101]	22.2	19.7	15.8	12.6	13.1
Richard *et al.* [102]	24.8	23.9	22.0	19.2	14.0
Zhu *et al.* [103]	-	-	-	-	14.9
Ours	**32.2**	**29.7**	**25.8**	**20.6**	**18.5**

TABLE VI
ACTION DETECTION RESULTS (%) ON THUMOS'14 WITH VARIED IOU THRESHOLD α. ALL PERFORMANCES ARE REPORTED USING MAP.

Model	0.1	0.2	0.3	0.4	0.5
Karaman *et al.* [4]	4.6	3.4	2.1	1.4	0.9
Sun *et al.* [104]	12.4	11.0	8.5	5.2	4.4
Wang *et al.* [3]	18.6	17.0	14.0	11.7	8.3
Oneata *et al.* [100]	36.6	33.6	27.0	20.8	14.4
Heilbron *et al.* [105]	36.1	32.9	25.7	18.2	13.5
Richard *et al.* [102]	39.7	36.7	30.0	23.2	15.2
Yeung *et al.* [75]	48.9	44.0	36.0	26.4	17.1
Zhu *et al.* [103]	47.7	43.6	36.2	28.9	19.0
Shou *et al.* [106]	47.7	43.5	36.3	28.7	19.0
Yuan *et al.* [107]	51.4	42.6	33.6	26.1	18.8
Xu *et al.* [108]	**54.5**	**51.5**	**44.8**	**35.6**	**28.9**
Ours	47.4	45.9	39.4	33.0	26.5

classifier over popular joint training manner, which is widely used in attention-based model [74] [75] [76], we train the same SVM classifier with ours and it along with our attention model in a joint manner (denoted as "joint"). The performance comparisons are presented in Table II. As can be seen, using an additional SVM is the best of all in terms of overall mAP performance on 20 classes of the ActivityNet "Sports" subset. Note that the right column is a naive implementation which simply employs the output of RNN for classification as original attention work [29].

Evaluation of training manner. We test two kinds of strategies to train our model. The first one is using an end-to-end manner, i.e. all the components are trained in a unified procedure like [75] [74], denoted as "unified-single/one-vs-all" which trains temporal bounding regressor and classifier simultaneously. Specifically, "unified-single" trains a single model while "unified-one-vs-all" does for each class. On the other hand, similarly, "separate-single/one-vs-all" trains temporal regressor and classifier separately. In particular, "separate-single" trains a single model for all classes while "separate-one-vs-all" does for each class. The detailed comparisons are shown in Table III. From the table we see a better mAP of "separate-one-vs-all" by 5% at least, which is used in our following experiments.

Given the investigations above, in the following comparison part, we use iDT to generate location candidates. The attention model is a 5-layer LSTM with 256 hidden units each and trained with an additional SVM for each class in "Separate-one-vs-all" manner.

D. Results and Comparison

MPII-Cooking. We compare our method to a sliding window baseline similar to [90] and [102]. Table V shows our method performs better than both the baseline and recent state-of-the-art approaches [101] [102] [103]. An interesting work is [102] where their novelty is that it includes a length and language model in addition to an action classifier. However, their runtime during inference is quadratic in the number of frames. By limiting the maximal action length to a constant, they can solve the action detection problem in a reasonable time, but this not easily scalable to long videos. When we compare our approach to [101], we report their performance by applying the implementation kindly provided by the authors. [103] develops a novel temporal actionness regression module that estimates what proportion of a clip contains action. In terms of mAP performance, our method outperforms all of these methods by 7% at least.

THUMOS'14. Table IV shows the per-class performance of our model on THUMOS'14 dataset using $\alpha = 0.5$. Comparison approaches are two top submissions [100] [3] on the THUMOS'14 leaderboard and a recent attention-based method [75]. Among all these four methods, our model outperforms on 14 out of 20 classes. Notably, it shows significant improvement on some of the most challenging classes in the dataset such as "Frisbee Catch" and "Hammer Throw". The model's ability to reason holistically on action extents enables it to infer temporal boundaries even when frame appearance is challenging: e.g. similar pose and environment, or abrupt scene change.

Next we report mAP for all the classes using different IOU thresholds, and compare with other approaches in Table VI. Most of these methods compute iDT and/or CNN features over temporal windows, and use a sliding window approach with non-maximum suppression to obtain predictions. This means that visual clues of all the frames have to be computed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2887061, IEEE Transactions on Circuits and Systems for Video Technology

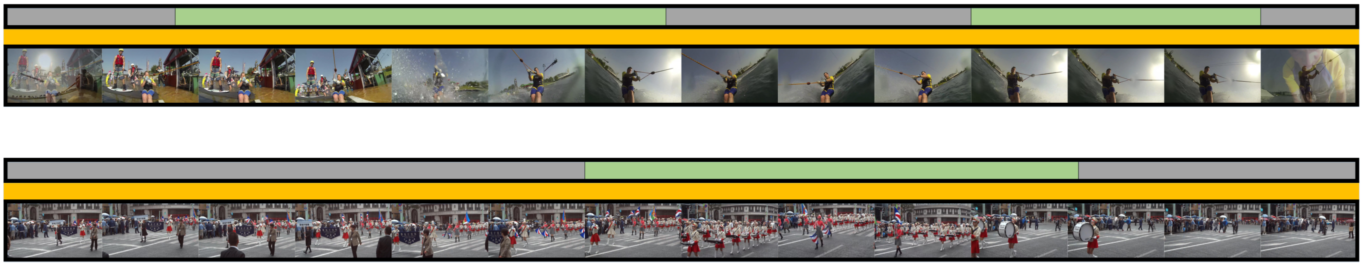


Fig. 8. Examples of predicted action instances on ActivityNet dataset. We select two long-range actions and test the performance of our method. Our model can sometimes produce more than one prediction about the temporal bound of action which is common in complex, untrimmed videos.

The three models in top section of Table VI use Fisher vector encoded iDT features on video clips and perform post-processing to refine the localization scores. [100] is the winning approach of the THUMOS'14 challenge. It uses video-level action classification scores as a context feature which greatly improves the performance. The five methods in middle section of Table VI are the most recent state-of-the-art results. Note that [106] and [103] outperform our results when $\alpha = 0.1$ and 0.2. This is perhaps due to the fact that their method concern more about fine-grained actions, since many ground truth instances in this dataset can be matched when α is small. We also see several recent works report competitive or superior than our method in certain cases [75] [103] [106] [107]. However, our result is competitive in the case of $\alpha = 0.5$. A recent work [108] that uses a region-C3D network performs best on this dataset.

ActivityNet. Table VII shows the action detection performance comparison on ActivityNet under different IOU thresholds α. The results of Heilbron et al. [93] are produced on their test set, which is not publicly available; therefore, their results are not directly comparable to ours. The middle four methods are from a recent work [94]. The LSTM models greatly outperform the CNN model. Their LSTM-s model trained with both the classification loss and ranking loss on the detection score. An alternative model called LSTM-m is trained with classification loss and ranking loss on the discriminative margin. Compared with their results, our model performs better when α is large which means the task is more difficult.

The following two results are submissions to ActivityNet Challenge 2016. [109] conducts detection task with the help of untrimmed classification. They generate the trimmed action proposals by combining frame-level binary classification with dynamic programming. Their results suggest that untrimmed video classification models can be used as stepping stone for temporal detection. [6] organizes videos with fixed 16-frame clips and then individually extracts both audio and visual features. Visual features were extracted from a pretrained C3D network, while MFCC coefficients were extracted for audio. Although using features of other media, their results are lower than ours when $\alpha = 0.5$. This indicates that our motion-centric model is not sensitive to temporal IOU. Very recently, several works including structured segment networks (SSN) [112], semantic context cascade (SCC) [111], and [110] provided competitive results, where the best performance in terms of $\alpha = 0.5$ reaches 43.2. However, they did not provide other results conditioned on various α.

E. Qualitative Analysis

In Fig. 8, we present a qualitative analysis of proposals generated by our class-induced method. Note the ability of our method to highly score proposals that are related with a previously seen action. For example, all the five best ranked proposals are related with one of the 20 classes on THUMOS'14 dataset. As illustrated in the figure, our proposal method is able to tightly localize the actions even that have a distance in the time line. Additionally, our method can often escape from unrelated region thanks to the motion-centric generator which produces a compact subset of location candidates in advance. Interestingly, we find an incomplete high jump action ranked in the bottom. This is evidence that our proposal method is able to discard low quality proposals.

Fig. 9 presents a detailed procedure of model performs given a video. Note that the input video shown in the figure spans 50 frames between consecutive shown frames. As can be seen, the attention model can jump on competitive candidates and finally achieve a reliable prediction which is colored in green.

F. Runtime Analysis

At last we report some runtime analysis of our method. We measure the processing speed of our method during inference and compare it with several recent competitive works. The reported time is the average FPS (frame per second) needed to generate the prediction for an average length video from THUMOS'14 (3minutes). As Table VIII shows, when comparing against full-processing methods such as action localization proposals from dense trajectories (APT) [101], action proposals from greedy search (APG) [113], binary proposal classifier (BPC) [105], Sparse-prop [105], BoFrag [114] and SCNN [106], we are able to obtain a speedup at least more than 1.5 times. Note that there are two recent works faster than our method. In particular, DAP [115] incorporates a proposal prediction step on top of LSTM and predicts at 134.1 FPS. R-C3D constructs the proposal and classification pipeline in an end-to-end fashion and these two stages share the features making it significantly faster, achieving 569 FPS. This results suggest that there is potential to further improve the speed of method by integrating location candidate generation with the attention model in an end-to-end manner.

TABLE VII
ACTIVITY DETECTION PERFORMANCE (%) MEASURED IN MAP AT DIFFERENT IOU THRESHOLDS α.

IOU Threshold(α) Method	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Heilbron *et al.* MF [93]	13.8	12.6	13.2	14.0	13.3	-	-	-
Heilbron *et al.* DF [93]	14.5	14.9	15.1	14.9	13.5	-	-	-
Heilbron *et al.* SF [93]	16.2	17.0	17.2	16.6	14.3	-	-	-
Heilbron *et al.* MF+DF+SF [93]	16.2	17.0	17.2	16.6	14.3	-	-	-
Ma *et al.* CNN [94]	30.1	26.9	23.4	21.2	18.9	17.6	16.5	15.8
Ma *et al.* LSTM [94]	48.1	44.3	46.3	35.6	31.3	28.3	26.0	24.6
Ma *et al.* LSTM-m [94]	52.6	48.9	45.1	40.1	35.1	31.8	29.1	27.2
Ma *et al.* LSTM-s [94]	**54.0**	**50.1**	**46.3**	41.2	36.4	33.0	30.4	28.7
UPC Submission [109]	-	-	-	-	22.5	-	-	-
Oxford Submission [6]	-	-	-	-	28.7	-	-	-
Singh *et al.* [110]	-	-	-	-	36.4	-	-	-
Caba *et al.* [111]	-	-	-	-	39.9	-	-	-
Zhao *et al.* [112]	-	-	-	-	**43.2**	-	-	-
Ours	46.3	43.8	42.7	**41.3**	38.0	**36.0**	**34.6**	**31.4**

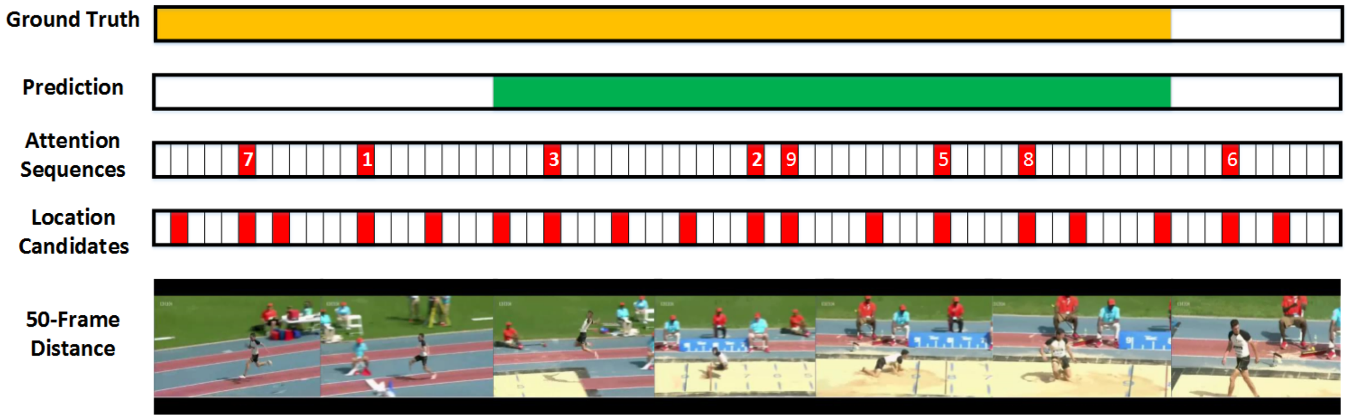


Fig. 9. An example of attention policy learning procedure on THUMOS'14 dataset. The generated location candidates and observed frames are colored in red. Our model jumps back and forth on the candidates and gradually determine where an action has high probabilities to occur. The temporal bound prediction is colored in green while the ground truth is colored in orange.

TABLE VIII
ACTION DETECTION SPEED DURING INFERENCE. WE REPORT THE AVERAGE TIME NEEDED FOR AN AVERAGE LENGTH VIDEO FROM THUMOS'14 (3MINUTES).

Method	FPS	Method	FPS
APT [101]	0.68	S-CNN [106]	60
BoFrag [114]	1.88	DAP [115]	134
Sparse-prop [105]	10.2	Region-C3D [108]	**569**
APG [113]	15	Ours (5-layer, 15-times)	96

V. CONCLUSIONS

In this paper, we introduce a motion-centric attention model for action detection in untrimmed videos. We show our model is able to yield fast and accurate predictions of temporal bound of action along with semantic label. Experiments on large-scale benchmarks show the effectiveness of our method in both accuracy and speed. Our potential future work is to integrate other pre-processing techniques such as saliency detection for attention model to produce more reliable predictions. Another future direction is to use deep networks for location generation to integrate the whole framework in an end-to-end manner.

ACKNOWLEDGMENT

This work was supported in part by the Shenzhen Peacock Plan (20130408-183003656), in part by the Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS-201703031405467), and in part by the National Natural Science Foundation of China (U-1613209).

REFERENCES

[1] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *ICCV*, 2013, pp. 3551–3558.

[2] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *NIPS*, 2014, pp. 568–576.

[3] L. Wang, Y. Qiao, and X. Tang, “Action recognition and detection by combining motion and appearance features,” *THUMOS14 Action Recognition Challenge*, vol. 1, p. 2, 2014.

[4] S. Karaman, L. Seidenari, and A. Del Bimbo, “Fast saliency based pooling of fisher encoded dense trajectories,” in *ECCV THUMOS Workshop*, vol. 1, 2014, p. 6.

[5] L. Wang, Z. Wang, Y. Xiong, and Y. Qiao, “Cuhk&siat submission for thumos15 action recognition challenge,” *THUMOS'15 Action Recognition Challenge*, vol. 3, no. 4, 2015.

[6] G. Singh and F. Cuzzolin, “Untrimmed video classification for activity detection: submission to activitynet challenge,” *arXiv preprint arXiv:1607.01979*, 2016.

[7] M. Raptis and L. Sigal, “Poselet key-framing: A model for human activity recognition,” in *CVPR*, 2013, pp. 2650–2657.

[8] L. Liu, L. Shao, and P. Rockett, "Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition," *Pattern recognition*, vol. 46, no. 7, pp. 1810–1818, 2013.

[9] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 810–822, 2014.

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: towards good practices for deep action recognition," in *ECCV*, 2016, pp. 20–36.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *TPAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[12] J. Wang, W. Wang, R. Wang, and W. Gao, "Learning class-specific pooling shapes for image classification," in *ICME*, 2015, pp. 1–6.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[14] J. Wang, W. Wang, R. Wang, and W. Gao, "Image classification using rbm to encode local descriptors with group sparse learning," in *ICIP*, 2015, pp. 912–916.

[15] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[16] J. Wang, W. Wang, R. Wang, and W. Gao, "A compact shot representation for video semantic indexing," in *ICIP*, 2015, pp. 3265–3269.

[17] Y. Zhang, W. Wang, and J. Wang, "Collaborative networks for person verification," in *ACM Multimedia Workshop on Multimedia Verification*, 2017, pp. 3–11.

[18] J. Wang, W. Wang, R. Wang, and W. Gao, "Csps: An adaptive pooling method for image classification," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1000–1010, 2016.

[19] Y. Zhang, W. Wang, and J. Wang, "Deep discriminative network with inception module for person re-identification," in *IEEE Visual Communications and Image Processing*, 2017, pp. 1–4.

[20] H. Song, W. Wang, J. Wang, and R. Wang, "Collaborative deep networks for pedestrian detection," in *IEEE International Conference on Multimedia Big Data*, 2017, pp. 146–153.

[21] Y. Zhang, W. Wang, and J. Wang, "Aligned local descriptors and hierarchical global features for person re-identification," in *Advances in Multimedia Information Processing*, 2017, pp. 418–427.

[22] J. Wang, W. Wang, R. Wang, and W. Gao, "Beyond monte carlo tree search: Playing go with deep alternative neural network and long-term evaluation," in *AAAI*, 2017, pp. 1576–1582.

[23] L. Itti, C. Koch, and E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*. IEEE Computer Society, 1998.

[24] D. L. Mayer and V. Dobson, "Visual acuity development in infants and young children, as assessed by operant preferential looking," *Vision research*, vol. 22, no. 9, pp. 1141–1151, 1982.

[25] A. L. Yarbus, "Eye movements during perception of complex objects," in *Eye movements and vision*, 1967, pp. 171–211.

[26] J. Jonides, D. E. Irwin, S. Yantis *et al.*, "Integrating visual information from successive fixations," *Science*, vol. 215, no. 4529, pp. 192–194, 1982.

[27] L. Itti, G. Rees, and J. K. Tsotsos, *Neurobiology of attention*. Academic Press, 2005.

[28] A. Frischen, A. P. Bayliss, and S. P. Tipper, "Gaze cueing of attention: visual attention, social cognition, and individual differences." *Psychological bulletin*, vol. 133, no. 4, p. 694, 2007.

[29] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *NIPS*, 2014, pp. 2204–2212.

[30] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *ICML*, 2015, pp. 1462–1471.

[31] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015, pp. 577–585.

[32] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[33] K. Xu, X. Jiang, and T. Sun, "Two-stream dictionary learning architecture for action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 567–576, 2017.

[34] X. Chen, J.-N. Hwang, D. Meng, K.-H. Lee, R. L. de Queiroz, and F.-M. Yeh, "A quality-of-content-based joint source and channel coding for human detections in a mobile surveillance cloud," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 19–31, 2017.

[35] Y. Xian, X. Rong, X. Yang, and Y. Tian, "Evaluation of low-level features for real-world surveillance event detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 624–634, 2017.

[36] S. Cho and H. Byun, "A space-time graph optimization approach based on maximum cliques for action detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 661–672, 2016.

[37] T.-F. Su, C.-K. Chiang, and S.-H. Lai, "A multiattribute sparse coding approach for action recognition from a single unknown viewpoint," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1476–1489, 2016.

[38] D. P. Barrett and J. M. Siskind, "Action recognition by time series of retinotopic appearance and motion features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 12, pp. 2250–2263, 2016.

[39] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[40] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[41] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, p. 28, 2015.

[42] I. Laptev, "On space-time interest points," *IJCV*, vol. 64, no. 2-3, pp. 107–123, 2005.

[43] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC*, 2008, pp. 1–10.

[44] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *ACM MM*, 2007, pp. 357–360.

[45] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009, pp. 1–11.

[46] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *CVPR*, 2010, pp. 2046–2053.

[47] A. Gilbert, J. Illingworth, and R. Bowden, "Fast realistic multi-action recognition using mined dense spatio-temporal features," in *ICCV*, 2009, pp. 925–931.

[48] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *CVPR*, 2009, pp. 1948–1955.

[49] F. Diego, J. Serrat, and A. M. López, "Joint spatio-temporal alignment of sequences," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1377–1387, 2013.

[50] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *BMVC*, 2012, pp. 30–1.

[51] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[52] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *CVPR*, 2012, pp. 1234–1241.

[53] L. Wang, Y. Qiao, and X. Tang, "Video action detection with relational dynamic-poselets," in *ECCV*, 2014, pp. 565–580.

[54] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu, "Action recognition with actons," in *CVPR*, 2013, pp. 3559–3566.

[55] M. Jain, J. V. Gemert, H. Jgou, P. Bouthemy, and C. G. M. Snoek, "Tubelets: Unsupervised action proposals from spatiotemporal supervoxels," *International Journal of Computer Vision*, no. 368, pp. 1–25, 2016.

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[57] J. Wang, W. Wang, and W. Gao, "Beyond knowledge distillation: Collaborative learning for bidirectional model assistance," *IEEE Access*, vol. 6, pp. 39 490–39 500, 2018.

[58] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014, pp. 1725–1732.

[59] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *TPAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[60] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015, pp. 4489–4497.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2887061, IEEE Transactions on Circuits and Systems for Video Technology

[61] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *ICCV*, 2015, pp. 4597–4605.

[62] J. Wang, W. Wang, R. Wang, W. Gao *et al.*, "Deep alternative neural network: Exploring contexts as early as possible for action recognition," in *NIPS*, 2016, pp. 811–819.

[63] Z. Li, W. Wang, N. Li, and J. Wang, "Tube convnets: Better exploiting motion for action recognition," in *ICIP*, 2016, pp. 3056–3060.

[64] X. Chen, W. Wang, and J. Wang, "Long-term video interpolation with bidirectional predictive network," in *VCIP*, 2017, pp. 1–4.

[65] J. Wang, W. Wang, and W. Gao, "Multiscale deep alternative neural network for large-scale video classification," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2578–2592, 2018.

[66] M. Jain, J. Gemert, C. G. Snoek *et al.*, "University of amsterdam at thumos challenge 2014," in *ECCV THUMOS Challenge*, 2014.

[67] A. Gorban, H. Idrees, Y. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," in *CVPR workshop*, 2015.

[68] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *ICCV*, 2015, pp. 2625–2634.

[69] X. Chen, W. Wang, J. Wang, and W. Li, "Learning object-centric transformation for video prediction," in *ACM Multimedia*, 2017, pp. 1503–1512.

[70] B. Chen, W. Wang, and J. Wang, "Video imagination from a single image with transformation generation," in *ACM Multimedia Thematic Workshops*, 2017, pp. 358–366.

[71] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[72] F. Wang and D. M. J. Tax, "Survey on the attention based RNN model and its applications in computer vision," *arXiv preprint arXiv:1601.06823*, 2016.

[73] T. V. Nguyen, Z. Song, and S. Yan, "Stap: Spatial-temporal attention-aware pooling for action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 77–86, 2015.

[74] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[75] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," pp. 2678–2687, 2016.

[76] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[77] W. Li, W. Wang, X. Chen, J. Wang, and G. Li, "A joint model for action localization and classification in untrimmed video with visual attention," in *ICME*, 2017, pp. 619–624.

[78] X. Chen, W. Wang, W. Li, and J. Wang, "Attention-based two-phase model for video action detection," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 81–93.

[79] Y. Yan, B. Ni, and X. Yang, "Predicting human interaction via relative attention model," *arXiv preprint arXiv:1705.09467*, 2017.

[80] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned spatio-temporal attention for human action recognition," *arXiv preprint arXiv:1703.10106*, 2017.

[81] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," *CVPR*, 2017.

[82] J. Wang, W. Wang, and W. Gao, "Predicting diverse future frames with local transformation-guided masking," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[83] J. Ba, R. R. Salakhutdinov, R. B. Grosse, and B. J. Frey, "Learning wake-sleep recurrent attention models," in *NIPS*, 2015, pp. 2593–2601.

[84] W. Sultani and I. Saleemi, "Human action recognition across datasets by foreground-weighted histogram decomposition," in *CVPR*, 2014, pp. 764–771.

[85] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *CVPR*, 2013, pp. 2555–2562.

[86] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[87] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei, "Detecting events and key actors in multi-person videos," in *CVPR*, 2016, pp. 3043–3053.

[88] M. Xin, H. Zhang, M. Sun, and D. Yuan, "Recurrent temporal sparse autoencoder for attention-based action recognition," in *Neural Networks (IJCNN), 2016 International Joint Conference on*, 2016, pp. 456–463.

[89] R. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[90] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *CVPR*, 2012, pp. 1194–1201.

[91] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[92] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[93] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015, pp. 961–970.

[94] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *CVPR*, 2016, pp. 1942–1950.

[95] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *arXiv preprint arXiv:1704.00389*, 2017.

[96] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[97] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008, pp. 1–8.

[98] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011, pp. 3169–3176.

[99] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *ICPR*, vol. 3, 2006, pp. 850–855.

[100] D. Oneata, J. Verbeek, and C. Schmid, "The lear submission at thumos 2014," 2014.

[101] J. Gemert, M. Jain, E. Gati, C. G. Snoek *et al.*, "Apt: Action localization proposals from dense trajectories," in *BMVC*, 2015, pp. 1–12.

[102] A. Richard and J. Gall, "Temporal action detection using a statistical language model," in *CVPR*, 2016, pp. 3131–3140.

[103] Y. Zhu and S. D. Newsam, "Efficient action detection in untrimmed videos via multi-task learning," in *IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 197–206.

[104] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, "Temporal localization of fine-grained actions in videos by domain transfer from web images," in *ACM Multimedia*, 2015, pp. 371–380.

[105] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *CVPR*, 2016, pp. 1914–1923.

[106] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016, pp. 1049–1058.

[107] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *CVPR*, 2016, pp. 3093–3102.

[108] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *ICCV*, 2017, pp. 5794–5803.

[109] A. Montes, A. Salvador, and X. Giro-i Nieto, "Temporal activity detection in untrimmed videos with recurrent neural networks," *arXiv preprint arXiv:1608.08128*, 2016.

[110] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *CVPR*, 2016, pp. 1961–1970.

[111] F. Caba Heilbron, W. Barrios, V. Escorcia, and B. Ghanem, "Scc: Semantic context cascade for efficient action detection," in *CVPR*, 2017, pp. 1454–1463.

[112] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," *arXiv preprint arXiv:1704.06228*, 2017.

[113] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *CVPR*, 2015, pp. 1302–1311.

[114] P. Mettes, J. C. V. Gemert, S. Cappallo, T. Mensink, and C. G. M. Snoek, "Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting," in *ICMR*, 2015, pp. 427–434.

[115] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *ECCV*, 2016, pp. 768–784.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2887061, IEEE Transactions on Circuits and Systems for Video Technology

Jinzhuo Wang received the B.S. degree from the School of Electronic Engineering and Computer Science, Peking University, in 2013. He is currently a Ph.D. candidate at the school of Electronic Engineering and Computer Science, Peking University. His research interests include computer vision and deep learning. He has published several papers on relevant conferences and journals, such as NIPS, ACM Multimedia, AAAI, ICME, ICIP, IEEE Transactions on multimedia and IEEE Transactions on Circuits and Systems for Video Technology.

Wenmin Wang (M'16) received the Ph.D. degrees in computer architecture from Harbin Institute of Technology, China, in 1989. After then, he worked as an assistant professor and associate professor, at Harbin University of Science and Technology as well as Harbin Institute of Technology. Since 1992, he gained about 18 years of oversea industrial experiences in Japan and America, in where served as staff engineer, chief engineer, general manager of software division, and etc. He came back the academia of China by the end of 2009, as a professor works at the School of Electronic and Computer Engineering, Peking University, China. His current research interests include computer vision, multimedia retrieval, artificial intelligence and machine learning.

Wen Gao (M'92-SM'05-F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991.

He is a Professor of computer science with Peking University, China. Before joining Peking University, he was a Professor of computer science with the Harbin Institute of Technology from 1991 to 1995, and a Professor with the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He served on the editorial board for several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, the *Eurasip Journal of Image Communications*, and the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.