Multi-Modality Multi-Task Recurrent Neural Network for Online Action Detection

Jiaying Liu, *Senior Member, IEEE*, Yanghao Li, *Student Member, IEEE*, Sijie Song, *Student Member, IEEE*, Junliang Xing, *Member, IEEE*, Cuiling Lan, *Member, IEEE*, and Wenjun Zeng, *Fellow, IEEE*

***Abstract*— Online action detection is a brand new challenge and plays a critical role in visual surveillance analytics. It goes one step further than a conventional action recognition task, which recognizes human actions from well-segmented clips. Online action detection is desired to identify the action type and localize action positions on the fly from the untrimmed stream data. In this paper, we propose a multi-modality multi-task recurrent neural network, which incorporates both *RGB* and *Skeleton* networks. We design different temporal modeling networks to capture specific characteristics from various modalities. Then, a deep long short-term memory subnetwork is utilized effectively to capture the complex long-range temporal dynamics, naturally avoiding the conventional sliding window design and thus ensuring high computational efficiency. Constrained by a multi-task objective function in the training phase, this network achieves superior detection performance and is capable of automatically localizing the start and end points of actions more accurately. Furthermore, embedding subtask of regression provides the ability to forecast the action prior to its occurrence. We evaluate the proposed method and several other methods in action detection and forecasting on the online action detection data set and gaming action data set datasets. Experimental results demonstrate that our model achieves the state-of-the-art performance on both tasks.**

***Index Terms*— Action detection, recurrent neural network, multi-modality, joint classification-regression.**

I. Introduction

HUMAN action detection is quite a challenge task in video surveillance analytics, which aims to not only recognize but also localize the actions in the video sequences. Different from action recognition and offline detection, which determine the action after completely observing an entire sequence, online action detection allows to detect the action type automatically as proceeded from the streaming videos. This technology makes it possible to forecast the start and end of the actions prior to their occurrence, which benefits a wide range of applications. For example, in home surveillance, some unexpected situations (*e.g.* falling down) can be alerted timely. It would also be appreciated for the smart-home system if it can predict the start of the impending action or the end of the ongoing actions, and then get something ready for the person it serves. Therefore, the detection and forecast system could respond to impending or ongoing actions accurately and as soon as possible, to make visual surveillance more intelligent.

In the past decades, many pioneer works on human action recognition and detection [1] have been investigated in RGB videos. Most of them [2]–[5] are designed for offline detection, which determines the action after the video is fully observed. The offline detection methods mainly adopt the sliding-window technique in [6]–[9], which divide the sequence into overlapped clips and the action recognition/classification is performed on each clip. For practical applications in video surveillance, it is much expected to intelligently localize the actions with uncertain length on the fly. Recently, an online action localization method [10] was proposed with representations at different granularity. Meanwhile, Recurrent Neural Networks (RNN) [11], [12] have shown superior performance on feature representation and temporal dynamics modeling on action recognition. Thus, researchers also investigated efficient action detection algorithms [13]–[15] that leverage the neural network for the untrimmed streaming data.

Moreover, biological observations indicate that skeleton, as a compact high level information of human representation, is valuable to recognize actions by humans [16]. Such representation is robust to the variation of illumination, backgrounds and viewpoints. With the prevalence of cost-effective depth cameras such as Microsoft Kinect and the advance of powerful human pose estimation technique [17], it is easier and more effective to obtain the depth data and further 3D skeleton data. Skeleton-based human representation has attracted more and more attention for action recognition [18]–[20] and action detection [21], [22]. Some traditional skeleton-based methods for action recognition and detection rely on hand-crafted descriptors [18], [23]–[28] from body joints. While recent works [19], [20], [29] have shown state-of-the-art results by exploiting RNN-based deep learning methods to automatically extract high-level representation and to model temporal dynamics at the same time, our previous work [30]

Manuscript received May 23, 2017; revised October 15, 2017 and November 28, 2017; accepted January 23, 2018. Date of publication January 30, 2018; date of current version September 4, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61772043 and Grant 61672519, in part by the Microsoft Research Asia under Grant FY17-RESTHEME-013, and in part by the CCF-Tencent Open Research Fund. This paper was recommended by Associate Editor W. Zuo.

J. Liu, Y. Li, and S. Song are with the Institute of Computer Science and Technology, Peking University, Beijing 100080, China (e-mail: liujiaying@pku.edu.cn; lyttonhao@pku.edu.cn; ssj940920@pku.edu.cn).

J. Xing is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: jlxing@nlpr.ia.ac.cn).

C. Lan and W. Zeng are with Microsoft Research Asia, Beijing 100080, China (e-mail: culan@microsoft.com; wezeng@microsoft.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2799968

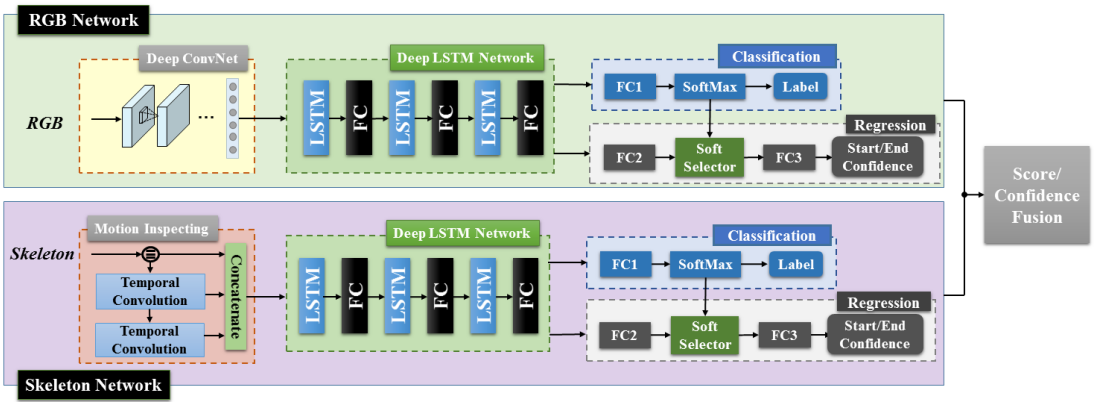


Fig. 1. Architecture of the proposed Multi-Modality Multi-Task RNN framework for online action detection and forecasting. Each small block corresponds to a specific layer. Specifically, FC (FC1, FC2, FC3) and LSTM represent fully-connected and Long Short-Term Memory (LSTM) layers, respectively. The Soft Selector layer is proposed to connect classification and regression tasks, which will be explained in Section IV-D.

went a further step to not only localize the start and end time of actions more accurately but also forecast their occurrence in advance.

Although skeleton data is an intrinsic high level representation, it is not always reliable due to the noise and occlusion. Without RGB, appearance information and spatial-temporal contexts are missing. It makes some actions look ambiguous when observing only skeleton data, especially in the case where the actor interacts with objects. Therefore, it is worth combining ***Multi-Modality*** data to make use of complementary information in them. The approach in [31] used additional 3D human-skeleton sequences to regularize the learning of Long Short-Term Memory (LSTM) and complemented poorly represented features in RGB videos. Although the skeleton sequences are independent with the RGB videos, they still provide helpful complementary information. In [32], RGBD-based Convolutional Neutral Network (CNN) features are used as spatio-temporal contexts to compensate for the noisy skeleton estimation with random forest. However, these methods are designed for action recognition and detection. There is a lack of a unified framework to integrate action detection and forecasting. And how to utilize multi-modal data to improve the performance of action forecasting is not well studied.

In this paper, we propose a Multi-Modality Multi-Task Recurrent Neural Network (MM-MT RNN) to accurately detect the actions and to localize the start and end positions of the actions on the fly from the streaming data. Fig. 1 shows the architecture of the proposed framework. For different input modalities, we design different temporal modeling networks to extract efficient temporal representations. Specifically, a *Deep Convolutional Network* is utilized for RGB data, while a *Motion Inspecting Layer* is exploited for skeleton data. Then, we use LSTM [33] to construct the recurrent layers to perform automatic feature learning and long-range temporal dynamics modeling. Our network is end-to-end trainable by optimizing a joint objective function of frame-wise action classification and temporal localization regression. On one hand, we perform frame-wise action classification to detect the actions timely. On the other hand, to better localize the start and end of actions, our network incorporates the regression of the start and end points of actions, and thus is capable to forecast their occurrences in advance based on the regressed curve. We train this classification and regression network jointly to obtain high detection accuracy. Note that the detection is performed frame-by-frame and the temporal information is automatically learned by the deep LSTM network without a sliding window design in a time efficient manner. Finally, with the help of joint RGB and skeleton data, we fuse the two-stream outputs as the final results. Experimental results on the Online Action Detection Dataset (OAD) and Gaming Action Dataset (G3D) datasets demonstrate the effectiveness of our method. This paper is an extension of our previous conference paper [30]. Based on the preliminary work, we introduce multi-modal data to compensate for the drawbacks of skeleton. At the same time, we propose different temporal modeling subnetworks to handle joint RGB and skeleton data, respectively. And we add the experimental analysis to evaluate the effectiveness of the proposed framework on OAD and G3D datasets. The main contributions of this paper are summarized as follows:

- To our best knowledge, we are the first to employ multi-modality data on the task of online action detection and forecasting. Our proposed method successfully leverages the advantages of multiple modalities by fusing them in the deep neural network framework.
- By the ablation analysis, we verify the effectiveness and necessity of each part in our proposed network, *i.e.* the Motion Inspecting layer, the Soft Selector module and the multi-task design scheme.
- Based on our collected large action dataset, we investigate the new problem of online action detection for streaming skeleton data by leveraging our multi-modality multi-task recurrent neural network. The experimental results also demonstrate the effectiveness of our method, which achieves state-of-the-art performance on action detection and forecasting.

The rest of this paper is organized as follows. In Section II, we review the related work on both RGB and skeleton-based action recognition and detection, respectively. In Section III, we formulate the online action detection problem. In Section IV, we illustrate the details of our

TABLE I
MULTI-MODALITY RESULTS ON THE NTU SUBSET IN ACCURACY (%)

Modal	Method	Accuracy (%)
Skeleton	LSTM	68.61
RGB	TSN [35]	69.56
Optical Flow	TSN [35]	80.55

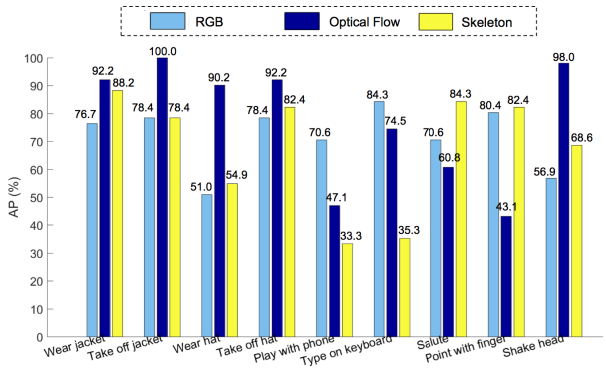


Fig. 2. Average precision on selected action types.

proposed method. Experimental results and discussions are presented in Section V. Finally, concluding remarks are given in Section VI.

II. MOTIVATION AND RELATED WORKS

A. Motivation

Different modal data describes actions from different perspectives and is capable of providing complementary information. Intuitively, skeleton joints and optical flow can describe human motion in a video sequence but ignore appearance information especially for human-object interactions. RGB videos are capable of conveying appearance information. To further exploit their strength and weakness of each modality, we conduct action recognition experiments on the NTU RGB+D dataset [34]. We adopt the state-of-the-art methods to evaluate the performance of each modality. For RGB and optical flow, we employ Temporal Segment Networks (TSN) [35]. For skeleton data, we build three stacked LSTM layers to test the performance. For all the experiments in this section, we randomly sample a subset from NTU datasets under cross view setting to avoid long-time training. This subset consists of 9065 videos, of which 6000 are used as training data and the rest is for testing.

Table I shows the baseline results. Skeleton and RGB provide comparable performance whenever skeleton data is sometimes unreliable due to the noisy joints caused by occlusion. Though optical flow lacks appearance information, it still gives the best results. On the one hand, optical flow provides pixel-level motion vectors, which provides finer granularity description of motion than that of skeleton. On the other hand, the classification of most actions can be achieved with only motion information. To further analyze the pros and cons of each modality, we select several typical action types and compare the average precision of each modal, which can be found in Fig. 2.

For RGB data, as in Fig. 2, it is usually confused for the sequential actions (*e.g.* wear/take off the jacket, wear/take off the hat). Based on the two-stream network, the classification on RGB data is obtained by fusing the probability of frames within one video. Thus, the temporal dynamics is not explicitly exploited. In contrast, actions with human-object interactions can be well classified by RGB data, even though the motion trajectories are similar (*e.g.* play the phone/type on the keyboard). In comparison, optical flow and skeleton can well handle the sequential actions thanks to the exploit of temporal information. However, the human-object interactions are ambiguous for them. In addition, optical flow is not good at classifications for motionless actions (*e.g.* saluting, pointing something with a finger). The action of saluting is mainly conveyed by the posture and little information can be obtained from optical flow. However, these action categories can be depicted and well recognized from skeleton data. For actions with tiny motions such as shaking head, optical flow shows better performance owing to its pixel-level motion description. From the analysis above, we conclude that each modal data has its advantages and the multi-modal data provide mutually complementary information. Therefore, to adapt various application scenarios and handle all kinds of action categories, it is worth leveraging multi-modal data towards good practice.

B. Related Works

1) RGB-Based Action Recognition and Detection: Action recognition and detection problem on RGB-based dataset have attracted a lot of research interests and been studied through the past decade. Traditional methods mainly focus on hand-crafting features for video representation. Bag-of-Feature (BOF) technique [36] tracks feature points in consecutive frames and encodes the extracted trajectories to capture the motion of objects in the video. This method was further improved by densely tracking points in the optical flow field with more features like Histogram of Oriented Gradient (HOG), Histogram of Flow (HOF) and Motion Boundary Histograms (MBH) and those encoded by Fisher Vector [37], [38]. But these carefully designed features is not robust enough compared to deep features learned from large scare training data. As a result, their performance has been outperformed by recent deep learning based methods.

Recently, deep learning has been exploited for action recognition [39], [40]. Instead of using hand-crafted features, deep approaches automatically learn robust feature representations directly from raw data and recognize actions synchronously. Besides a large number of CNNs [41]–[43] proven successful in modeling the image data, many novel deep models especially designed for video dynamic representations have been proposed recently. For example, VLAD3 [44] and Rank Pooling [45], [46] encode video representations based on the features extracted from CNN models. However, these methods only use CNN models as feature extractors without end-to-end design to extract better feature representations. Similar to many sequence modeling tasks [47], RNNs have also been exploited to model temporal dynamics for action recognition. In [11] and [12], CNN layers are constructed to extract visual

features while the followed recurrent layers are applied to handle temporal dynamics.

For action detection, existing methods mainly utilize either sliding-window scheme [6], [7], [21], [48], or action proposal approaches [49]–[52]. Due to a large number of action proposals to classify, these methods usually have redundant computation or unsatisfactory localization accuracy because of the overlapping design and unsupervised localization approach. Besides, it is not easy to determine the sliding-window size. Most methods are designed for offline action detection [8], [21], [48], [53]. Nevertheless, in many applications it is expected to recognize the action on the fly before the completion of the action, *e.g.*, to respond as fast as possible in human computer interaction. In [6], a learning formulation based on a structural SVM is proposed to recognize partial events, enabling early detection. To reduce the observational latency of human action recognition, a non-parametric moving pose framework [23] and a dynamic integral bag-of-words approach [54] are proposed respectively to detect actions earlier. Our model goes beyond early detection. Besides providing frame-wise class information, it forecasts the occurrence of start and end of actions before an action is performed in practical.

Our framework aims to address the online action detection in order to predict the action at each time slot efficiently without a sliding window design. We regress the start/end points in a supervised manner during the training, enabling a more accurate localization. Furthermore, it forecasts the start of the impending or end of the ongoing actions.

2) Skeleton-Based Action Recognition and Detection: For skeleton-based action recognition, many generative models [55], [56] such as Hidden Markov Models (HMM) were proposed to model temporal dynamics. Their disadvantages include the difficulty in estimating model parameters and learning them efficiently. Leveraging the insights from RGB-based action recognition, many discriminative approaches have been proposed with superior performance. There methods capture local features from the sequences and then are connected to typical classifiers. For instance, rotations and translations are used to represent geometric relationships of body parts in a Lie group [24], [25], and then Fourier Temporal Pyramids (FTP) or Dynamic Time Warping (DTW) are employed to temporally align the sequences and to model temporal dynamics. In [57], the covariance matrix is calculated to learn the co-occurrence of skeleton points. Furthermore, many methods [58]–[60] divide the human body into several parts and learn the co-occurrence information, respectively. A Moving Pose descriptor [23] is proposed to mine key frames temporally via a k-NN approach in both pose and atomic motion features.

Due to the design of specific handcrafted features, most methods mentioned above are limited in modeling temporal dynamics within a window having a certain length. Recently, deep learning methods are proposed to learn robust feature representations and to model the temporal dynamics for the whole video without segmentation for skeleton data. In [19], a hierarchical RNN is utilized to model the temporal dynamics for skeleton based action recognition. To exploit the inherent co-occurrences of skeleton joints, Zhu *et al.* [20] proposed a deep LSTM network to model the inherent correlations among skeleton joints in various actions. Liu *et al.* [61] further extended the RNN to spatial-temporal domains to analyze the hidden sources of action-related information. By converting skeleton sequences into images where the spatio-temporal information is reelected in the image, some CNN-based methods [62], [63] are proposed to model transformed skeleton images by convolutional layers. The above action recognition methods could not been directly applied to action detection task, since they assume there are only one action instance in the input video.

For skeleton-based action detection, several approaches [21], [30], [64] were proposed. To extract efficient feature representations, a dynamic bag of features [64] and a multi-scale feature descriptor [21] were proposed. Our previous paper [30] introduced a Joint Classification Regression RNN to avoid sliding window design which shows state-of-the-art performance for online action detection. In this work, we propose a Multi-Modality Multi-Task Recurrent Neural Network to exploit the advantages of both skeleton and RGB compensatingly. Different temporal modeling networks are designed for different modalities, respectively, which are further jointly modeled by a unified deep recurrent neural network.

3) Summary of Related Works: We summarize the above skeleton-based and RGB-based related works in Table II based on different tasks (action recognition and detection) and method category (traditional / deep learning methods). It should be noted that currently, most state-of-the-art methods leverage deep learning models like CNN or RNN for action analysis. In the bottom of the table, we also categorize several works for multi-modality action recognition. In [66], RGB

TABLE II
SUMMARY OF RELATED WORKS FOR ACTION ANALYSIS

Modality	Task	Category	Methods
RGB	Recognition	Traditional	BOF [36], IDT [38]
		Deep-based	VLAD3 [44] Rank Pooling [45], [46] Two-Stream [39], TSN [35] RNN-based [65], [66]
	Detection	Traditional	Sliding-window [6], [48] Action Proposal [50], [51] Early-detection [54]
		Deep-based	Sliding-window [7] Action Proposal [49], [52]
Skeleton	Recognition	Traditional	Generative model [55], [56] Moving Pose [23] Co-occurrence [58]–[60]
		Deep-based	RNN-based [19], [20], [29], [61] CNN-based [62], [63]
	Detection	Traditional	Sliding-window [21] Random-Forest [32]
		Deep-based	RNN-based [30]
Multi-modal	Recognition	Traditional	RGB + Audio [67]
		Deep-based	RGB + Depth [68], [69] Skeleton-regularizing [31]

and audio features are extracted respectively to take advantage of multi-modal features of movie actions. To achieve additional depth information, [67], [68] utilized depth frames to compensate visual RGB frames by training modal-specific deep models for gesture recognition. Different modalities are usually trained separately and then fused by probability scores [66]–[68]. The approach in [31] used additional independent 3D human-skeleton sequences to regularize the learning of LSTM network which is used for action recognition for RGB videos. Due to the fundamental differences in our problem, we cannot use these methods. Our method focuses on the online action detection and forecasting problem using multiple modalities. Specifically, we directly exploit the advantages of paired RGB and skeleton sequences, which could be captured Kinect devices.

III. Problem Formulation

In this section, we formulate the online action detection problem. To clarify the differences, offline action detection is first illustrated.

A. Offline Action Detection

Given a video observation $V = \{v_0, \ldots, v_{N-1}\}$ composed of frames from time 0 to $N-1$, the goal of action detection is to determine whether a frame v_t at time t belongs to an action among the predefined M action classes.

Without loss of generality, the target classes for the frame v_t are denoted by a label vector $\mathbf{y}_t \in R^{1\times(M+1)}$, where $y_{t,j} = 1$ means the presence of an action of class j at frame t and $y_{t,j} = 0$ means absence of this action. Besides the M classes of actions, a blank class is added to represent the situation in which the current frame does not belong to any predefined actions. Since the entire sequence is known, the determination of the classes at each time slot is to maximize the posterior probability

$$\mathbf{y}_t^* = \underset{\mathbf{y}_t}{\operatorname{argmax}}\, P(\mathbf{y}_t|V), \tag{1}$$

where $\mathbf{y}_t$ is the possible action label vector for frame v_t. Therefore, conditioned on the entire sequence V, the action label with the maximum probability $P(\mathbf{y}_t|V)$ is chosen to be the status of frame v_t in the sequence.

According to the action label of each frame, an occurring action i can be represented in the form $d_i = \{g_i, t_{i,start}, t_{i,end}\}$, where g_i denotes the class type of the action i, $t_{i,start}$ and $t_{i,end}$ correspond to the starting and ending time of the action, respectively.

B. Online Action Detection

In contrast to offline action detection, which relies on the whole video to make decisions, online detection determines which action the current frame belongs to without using future information. Thus, the method is capable to automatically estimate the start time and status of the current action. The problem can be formulated as

$$\mathbf{y}_t^* = \underset{\mathbf{y}_t}{\operatorname{argmax}}\, P(\mathbf{y}_t|v_0, \ldots, v_t). \tag{2}$$

Besides determining the action label, an online action detection system for streaming data is also expected to predict the starting and ending time of an action. Specially, a common expectation is to be aware of the occurrence of the action as early as possible and be able to predict the end of the action. To avoid some trivial situations (*e.g.* forecast too early without any evidences), we define an expected forecasting time T. For example, for an action $d_i = \{g_i, t_{i,start}, t_{i,end}\}$, the system is expected to forecast the start and end of the action during $[t_{i,start} - T, t_{i,start}]$ and $[t_{i,end} - T, t_{i,end}]$, respectively, ahead its occurrence. We define the optimization problem as:

$$(\mathbf{y}_t^*, \mathbf{a}_t^*, \mathbf{b}_t^*) = \underset{\mathbf{y}_t, \mathbf{a}_t, \mathbf{b}_t}{\operatorname{argmax}}\, P(\mathbf{y}_t, \mathbf{a}_t, \mathbf{b}_t|v_0, \ldots, v_t), \tag{3}$$

where $\mathbf{a}_t$ and $\mathbf{b}_t$ are two vectors, denoting whether actions are to start or to stop within the following T frames, respectively. For example, $a_{t,g_i} = 1$ means that the action of class g_i will start within T frames.

IV. Multi-Modality Multi-Task RNN for Online Action Detection

We propose an end-to-end trainable Multi-Modality Multi-Task Recurrent Neural Network to address the online action detection problem. Fig. 1 illustrates its general architecture, consisting of temporal modeling networks, classification subnetworks and regression subnetworks. Note that we construct different structures of temporal modeling networks to extract deep spatial and temporal features for different modalities, respectively. After extracting long-term dynamic features, the followed classification subnetworks and regression subnetworks share the same structures but different weights among different modalities. In the training, we first train the classification network, including the temporal modeling network and classification subnetwork, for the frame-wise action classification. Then, with the guidance of classification results through the Soft Selector, we jointly train the regressor and classifier to obtain more accurate localization of the start and end points.

In the following, we first briefly review the RNN and LSTM. Then, we introduce our proposed MM-MT RNN for online action detection.

A. Overview of RNN and LSTM

In contrast to traditional feed-forward neural networks, RNNs have self-connected recurrent connections which model the temporal evolution. The output response $\mathbf{h}_t$ of a recurrent hidden layer can be formulated as follows [69]

$$\mathbf{h}_t = \theta_h(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h), \tag{4}$$

where $\mathbf{W}_{xh}$ and $\mathbf{W}_{hh}$ are mapping matrices from the current inputs $\mathbf{x}_t$ to the hidden layer h and the hidden layer to itself. $\mathbf{b}_h$ denotes the bias vector. θ_h is the activation function in the hidden layer.

The above RNNs have difficulty in learning long range dependencies [70] due to vanishing gradient effect. To overcome this limitation, recurrent neural networks using LSTM [19], [33], [69] have been designed to mitigate the

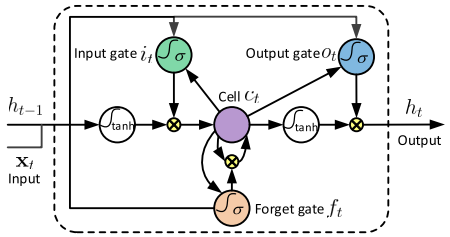


Fig. 3. The structure of an LSTM neuron, which contains an input gate i_t, a forget gate f_t, and an output gate i_t. Information is saved in the memory cell c_t.

vanishing gradient problem and to learn the long-range contextual information of a temporal sequence. Fig. 3 illustrates a typical LSTM neuron. In addition to a hidden output h_t, an LSTM neuron contains an input gate i_t, a forget gate f_t, a memory cell c_t, and an output gate o_t. At each time step, it can choose to read, write or reset the memory cell through the three gates. This strategy allows LSTM to memorize and to access information many time steps ago.

B. Temporal Modeling Networks for Different Modalities

Different input modalities have different forms and dimensions. For example, RGB frame is usually represented by high dimensional features, which contains human and background together, while the skeleton only contains dozens of high-level 3D human representation points. Thus, we design different temporal modeling networks for different modalities to learn spatial and temporal action representations in bottom layers of the network. Specifically, RGB data (*i.e.* raw RGB frame or optical flow between neighbor frames) is fed into a classic convolutional network (*e.g.* VGG [71] and Inception-BN [72]) to extract high-level semantic features. For the skeleton data, we propose a specific Motion Inspecting layer to incorporate short-term dynamics into raw skeleton points. Specifically, the module of feature extraction is followed by three LSTM layers and three non-linear fully-connected (FC) layers, which enables our model to handle long-term dynamics modeling with powerful learning capability.

1) Deep Convolutional Network for RGB Data: Deep Convolutional Networks come with great modeling capacity of learning discriminative representation from raw visual data. In our model, we also exploit convolutional networks to extract frame-wise spatial and temporal information from RGB data. Similar to the two-stream ConvNets [39] for action recognition, we study two kinds of the input RGB modalities, namely raw RGB frame and optical flow fields. A single RGB frame usually encodes static appearance at a specific time while the optical flow field focuses on capturing the motion information between two adjacent frames. After being fed into classic deep convolutional networks, both modalities can be represented as high-level semantic features in the top layers of deep convolutional networks (*e.g.* the flatten layer of Inception-BN [72] network). Then, a followed stacked LSTM network is responsible for extracting long-term dynamics based on the representative features from convolutional networks. Thus, this convolutional LSTM structure can successfully extract both spatial structure and temporal dynamics.

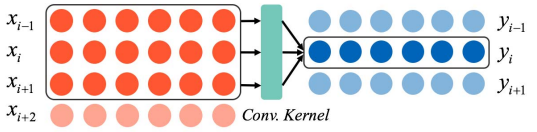


Fig. 4. The structure of the Motion Inspecting (MI) layer, where x_i and y_i correspond to the input and output data at time i.

2) Motion Inspecting Layer for Skeleton Data: Existing skeleton-based deep methods [19], [20], [30] only feed location information to the neural networks for action recognition or detection. However, in this way, it is difficult for the networks to automatically learn the high order information that encodes movements compactly. For example, the velocity and acceleration of each joint are effective discriminative statistics for action recognition. More specifically, the direction of the velocity at the joint helps distinguish the actions like *pushing* vs. *pulling*. And in the case of *punching* vs. *pushing*, where the joints share similar velocities, their accelerations further help differentiate one from the other.

Following the motivations described above, we introduce the Motion Inspecting (MI) layer to explicitly capture these high order derivatives, which largely boost the learning capacity of the whole model. It effectively extracts these derivatives for augmentation automatically and can be implemented by a temporal convolutional operator, which executes convolutions over the time steps. Furthermore, the involvement of high order information can finally influence the learning from the raw material and force the model to learn the velocity related features.

Specifically, for a given skeleton sequence $\{x_i\}$, we have a kernel $\mathbf{k} \in \mathbb{R}^{ks}$ as the learning parameters to extracts high order derivatives. The output y_i of this layer can be calculated as

$$y_i = \sum_{j=0}^{ks-1} \left(x_{i-j} \cdot k_j\right), \tag{5}$$

where ks is the kernel size. An example of MI layer with kernel size 3 is shown in Fig. 4.

In our experiments, we stack two convolution operations for further utilizing the input data and extracting richer statistics as shown in Fig. 1. This kind of production will generate a local measurement of the movement. We use zero-padding, to fill the remaining spaces at the beginning of the time where the product is invalid with zeros to align the temporal dimension. The MI layers, which focus on the short-term dynamics modeling, are then combined with the LSTM layers to construct the whole temporal modeling network for the skeleton branch.

C. Subnetwork for Classification Task

In the training, we first train a classification subnetwork together with the corresponding temporal modeling layers for frame-wise action recognition. The structure of this classification subnetwork is also shown in Fig. 1. The frame for each

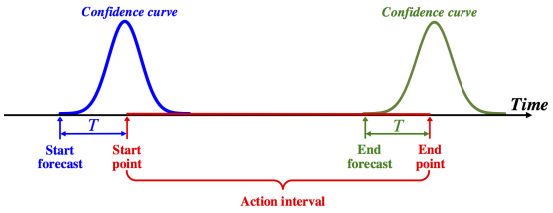


Fig. 5. Illustration of the confidence values around the start point and end point, which follows Gaussian-like curves with the confidence value 1 at the start and end point. (This figure is best viewed in color.)

modality first goes through the temporal modeling network, which is responsible for modeling the spatial structure and temporal dynamics. Then, a fully-connected layer FC1 and a SoftMax layer are added for the classification of the current frame. The output of the SoftMax layer is the probability distribution of the action classes $\mathbf{y}_t$. Following the problem formulation as described in Sec. III, the objective function of this classification task is to minimize the cross-entropy loss function

$$\mathcal{L}_c(V) = -\frac{1}{N}\sum_{t=0}^{N-1}\sum_{k=0}^{M} z_{t,k} \ln P(y_{t,k}|v_0, \ldots v_t), \tag{6}$$

where $z_{t,k}$ corresponds to the ground-truth label of frame v_t for class k, $z_{t,k} = 1$ means that the ground-truth class is the *k-th* class, and $P(y_{t,k}|v_0, \ldots, v_t)$ denotes the estimated probability of being the *k-th* action class of the frame v_t.

We train this network with Back Propagation Through Time (BPTT) [73] and use stochastic gradient descent with momentum to compute derivatives of the objective function with respect to all parameters. To prevent over-fitting, we utilize the dropout in three fully-connected layers.

D. Joint Classification and Regression

We fine-tune this network on the initialized classification model by jointly optimizing the classification and regression. Inspired by Joint Classification-Regression models used in Random Forest [74], [75] for other tasks (*e.g.* segmentation [74] and object detection [75]), we propose our joint learning to simultaneously make frame-wise classification, localize the start and end time points of actions, and to forecast them.

We define a confidence factor for each frame to measure the possibility of the current frame to be the start or end points of some actions. To better localize the start or end point, we use a Gaussian-like curve to describe the confidence, which centralizes at the actual start (or end) point as illustrated in Fig. 5. Taking the start point as an example, the confidence of the frame v_t with respect to the start point of action j is defined as

$$c_t^s = e^{-(t-s_j)^2/2\sigma^2}, \tag{7}$$

where s_j is the start point of the nearest (along time) action j to the frame v_t, and σ is the parameter to control the shape of the confidence curve. Note that at the start point time, *i.e.*, $t = s_j$, the confidence value is 1. Similarly, we denote the confidence of being the end point of one action as c_t^e. For the Gaussian-like curve, a lower confidence value suggests that the current frame has a larger distance to the start point and the peak point indicates the start point.

Such a design models the localization of start and end points into a soft formulation, preserves compatibility to the original localization problem and makes it possible to forecast. For localization, the start/end points are detected by checking the regressed peak points. For forecasting, the start (or end) of actions is acquired according to the current confidence response. We set a confidence threshold θ_s (or θ_e) according to the sensitivity requirement of the system to predict the start (or end) point. When the current confidence value is larger than θ_s (or θ_e), we consider that one action may start (or end) soon. Generally, a larger threshold corresponds to a later response but a more accurate forecast.

Using the confidence as the target value, we include this regression problem as another task in our RNN model, as shown in the bottom part of Fig. 1. This regression subnetwork consists of a non-linear fully-connected layer FC2, a Soft Selector layer, and a non-linear fully-connected layer FC3. Since we regress one type of confidence values for all the start points of different actions, we need to use the output of the action classification to guide the regression task. Therefore, we design a Soft Selector module to generate more specific features by fusing the output of SoftMax layer that describes the probabilities of classification together with the output of the FC2 layer.

We achieve that by using class specific element-wise multiplication of the outputs of SoftMax and FC2 layer. The information from the SoftMax layer for the classification task plays the role of class-based feature selection over the output features of FC2 for the regression task. A simplified illustration about the Soft Selector model is shown in Fig. 6. Assume that we have five action classes and the dimension of the FC2 layer output is reshaped to 7×5. The vector (marked by circles) with the dimension of 5 from the SoftMax output denotes the probabilities of the current frame belonging to five classes respectively. Element-wise multiplication is performed for each row of features. Then, integrating the SoftMax output plays the role of feature selection for different classes.

The final objective function of the Joint Classification-Regression is formulated as

$$\begin{aligned}\mathcal{L}(V) &= \mathcal{L}_c(V) + \lambda\mathcal{L}_r(V) \\ &= -\frac{1}{N}\sum_{t=0}^{N-1}\left[\left(\sum_{k=0}^{M} z_{t,k} \ln P(y_{t,k}|v_0, \ldots v_t)\right)\right. \\ &\quad \left. + \lambda \cdot \left(\ell(c_t^s, p_t^s) + \ell(c_t^e, p_t^e)\right)\right],\end{aligned} \tag{8}$$

where p_t^s and p_t^e are the predicted confidence values of start and end points, and λ is the weight for the regression task, ℓ is the regression loss function, which is defined as $\ell(x, y) = (x - y)^2$. In the training, the overall loss is a summarization of the loss from each frame v_t $(0 \le t < N)$. For a frame v_t, its loss consists of the classification loss represented by the

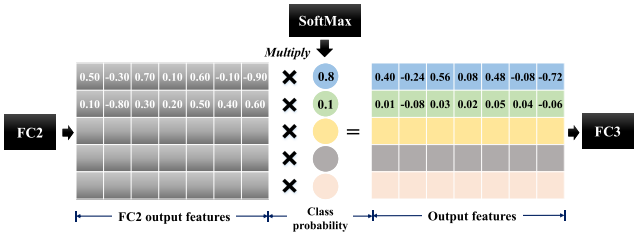


Fig. 6. Soft Selector for the fusion of classification output and features from FC2. Element-wise multiplication is performed for each row of features (we only show the first two rows here).

TABLE III
SUMMARY OF G3D AND OAD DATASETS

Datasets	G3D	OAD
Classes	20	10
Action Type	Game Actions	Daily Actions
Subjects	10	12
Videos	209	59
Actions per Video	3.0	14.2
Length per Video (frames)	342.6	3269.1
Length per Action (frames)	27.2	35.5

cross-entropy for the $M+1$ classes and the regression loss for identifying the start and end of the nearest action.

We fine-tune this entire network over the initialized classification model by minimizing the object function of the joint classification and regression optimization. To enable the classification result indicating which action will begin soon, we set the ground-truth label $z^s_{t,k} = 1$ in the training where $t_{k,start} - T \leq t < t_{k,start}$ for all actions, according to the expected forecast-forward value T as defined in Sec. III. Then, for each frame, the classification output indicates the impending or ongoing action class while two confidence outputs show the probability to be the start or end point. We set the peak positions of confidences to be the predicted action start (or end) time. Note that since the classification and regression results of the current frame are correlated with the current input and the previous memorized information of the LSTM network, the system does not need explicitly looking back, which draws a sharp distinction between the proposed method and sliding window design.

V. EXPERIMENTS

In this section, we evaluate the detection and forecast performance of the proposed method on two different RGB-D datasets, including Gaming Action dataset (G3D) [76] and a new collected online streaming dataset (OAD) proposed in our previous work [30]. We first adopt several ablation analysis to verify different components of model design. Then, we evaluate our proposed method and other compared methods with different evaluation protocols for action detection and forecasting, respectively.

A. Evaluation Datasets

Most published RGB-D datasets are generated for the classification task where actions are already pre-segmented [77], [78]. For online action detection, besides the video clips of a given action, we still need the video clips before the start point and those after the end point as well as the related timestamps. Many existing action detection datasets like THUMOS14 [79] and ActivityNet [80] do not contain skeleton and depth data, which are not suitable for our multi-modality setting. Thus, besides using an existing skeleton-based detection dataset G3D [76], we adopt a new online streaming dataset [30] that follows similar configurations to previous action recognition datasets, keeps compatible to action detection and serves for the online action detection problem.

1) Gaming Action Dataset (G3D): The G3D dataset contains 20 gaming actions captured by Kinect, grouped into seven categories, such as fighting, tennis, golf, tennis, bowling, FPS and driving. Each category also contains several different actions, *e.g.* right punch and left punch for Fighting category. The actions are performed by actors imitatively in the fixed shot environment. The various actions with different length and types make it suitable to evaluate the detection performance. At the same time, this dataset is also limited in some aspects. For example, the number and occurrence order of actions in the videos are unchanged and some actors seem motionless in the intervals of different actions, which make the dataset far away from real-life scenarios. It is also observed that, there are many RGB frames missing in this dataset. The missing problem degrades the performance a lot when RGB data is used.

2) Online Action Detection Dataset (OAD): This is a newly collected action dataset with long sequences for the online action detection problem. The dataset is captured using the Kinect v2 sensor, which collects color images, depth images and human skeleton joints synchronously. It is shot in a daily-life indoor environment. Different actors freely perform 10 actions, such as *drinking, writing, washing hands*. These daily actions are performed continuously (with idle time) in a single video. At the same time, different actions may have various duration length which makes the online action detection and forecasting more challenging. For example, *opening microwave* is a quick action while *sweeping* is a long-time action which takes uncertain length. It collects 59 long video sequences (in total 103,347 frames of 216 minutes).

A summary table of these two datasets is shown in Table III. Note that since the Kinect v2 sensor is capable of providing more accurate depth, OAD dataset [30] has more accurate tracked skeleton positions compared to previous skeleton datasets. In addition, different from G3D where videos in each type of category share the same actions, the acting orders and duration of the actions are arbitrary, which is closer to the real-life scenarios. At the same time, the average number of actions per video and the averaged length per video in OAD are also longer than those in G3D. The length of each sequence is long up to about 480 seconds and there are variable idle periods between different actions. These properties and configurations make OAD meet the requirements of realistic online action detection from streaming videos.

B. Parameter Settings and Compared Methods

For the train/test splitting settings, we follow the same schemes in [30] and [76] for OAD and G3D datasets, respectively. For the skeleton data, normalization processing on each skeleton frame is performed to be invariant to position, which is commonly adopted in previous works [20], [30]. Note that we do not perform sequential downsampling on the temporal dimension of videos like [19], [20], since we need to localize accurate start/end points for each action.

1) Network and Parameter Settings: We show the architecture of our network in Fig. 1. For the bottom temporal modeling networks, we use a Deep Convolutional Network for RGB data and a Motion Inspecting layer for the skeleton data. Similar to [35], we use the Inception-BN [72] structure as the bottom layers of RGB branch. Specifically, we use the flatten layer of Inception-BN and then feed it into the LSTM layers. For the optical flow modality, we discretize optical flow field into the interval from 0 to 255 by a linear transformation [35] to make the range of optical flow to be the same with RGB images. Then we use the same network structure as that for RGB frames. Since the scale of two detection datasets is relatively small to train the large Inception-BN network, we initialize our network for both RGB and optical flow with the models in [35], which pre-trained on the UCF-101 [81], a large action recognition dataset. During training, the weights before the flatten layer are fixed to prevent over-fitting. For the skeleton data, the kernel size of the temporal convolution operators in the MI layer is set as 3.

The number of neurons in the LSTM networks for both RGB and skeleton are 100, 100, 110, 110, 100, 100 for the six layers respectively, including three LSTM layers and three FC layers. The number of neurons in the FC1 layer corresponds to the number of action classes $M+1$ and the number of neurons in the FC2 layer is set to $10\times(M+1)$. For the FC3 layer, there are two neurons corresponding to the start and end confidences, respectively. The forecast response threshold T can be set based on the requirement of the applications. In this paper, we set $T = 10$ (around one second) for the following experiments. The parameter σ in (7) is set to 5. λ is defined to balance the weight of classification and regression tasks in the final loss function (8). Since the regression task is guided by the classification task using Soft Selector layer. We first set λ to be 0 to initialize the classification subnetwork and then increase it gradually from 0 to 10 during the fine-tuning of the entire network. Note that we use the same parameter settings for both OAD and G3D datasets.

We follow the same dataset split settings in [30] during training. For OAD dataset, we use 30 sequences for training, 20 sequences for testing and 9 sequences as the validation set. The 9 long videos are also used for the evaluation of the running speed. For the G3D dataset, we use the same setting as used in [76], i.e. 20 videos as training set and 10 videos for testing for each action category.

2) Compared Methods: To verify the effectiveness of the proposed MM-MT RNN, we compare several different methods on the action detection and forecasting tasks. First, we have implemented two baselines with the sliding window scheme for comparison. (a) SVM-SW. We train an SVM detector to detect the action with sliding window design (SW). (b) RNN-SW. The baseline method Deep LSTM in [20] achieves good results on many skeleton-based action recognition datasets. We train the classifiers and perform the detection based on sliding window design. We set the window size to 10 with the step of 5 for both RNN-SW and SVM-SW. We experimentally evaluate different window sizes and find 10 giving relatively good average performance. We also compare two other methods. (c) JCR-RNN [30]. It is the model proposed in our previous conference paper [30]. (d) RF+ST [32]. A random forest (RF)-based action detection method uses spatial-temporal contexts in RFs to improve the action detection performance with both skeleton and RGB data. (e) SM-MT RNN. A simplified version of our model uses only skeleton modality data with the specific network for skeleton branch. We denote it as Single-Modality Multi-Task RNN model (SM-MT RNN).

C. Evaluation in Action Detection

In this section, we evaluate the proposed method in action detection task. First, we introduce the adopted evaluation criteria for this task. Then, we conduct several ablation experiments to validate the effectiveness of the multi-modality design and specific proposed modules. We also compare the results with several baseline methods on the two action detection datasets.

1) Evaluation Criteria: We use three different evaluation protocols to measure the detection results.

- $F1$-Score. Similar to the protocol used in object detection from images [82], we define a criterion to determine a correct detection. A detection is correct when the overlapping ratio α between the predicted action interval I and the ground-truth interval I^* exceeds a threshold, *e.g.*, 60%. α is defined as,

$$\alpha = \frac{|I \cap I^*|}{|I \cup I^*|}, \tag{9}$$

  where $I \cap I^*$ denotes the intersection of the predicted and ground-truth intervals and $I \cup I^*$ denotes their union. With the above criterion to determine a correct detection, the $F1$-Score is defined as,

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}. \tag{10}$$

- SL-Score. To evaluate the accuracy of the localization of the start point for an action, we define a Start Localization Score (SL-Score) based on the relative distance between the predicted and the ground-truth start time. Suppose that the detector predicts that an action will start at time t and the corresponding ground-truth action interval is $[t_{start}, t_{end}]$, the score is calculated as $\exp\{-|t - t_{start}|/(t_{end} - t_{start})\}$. For false positive or false negative samples, the score is set to 0.
- EL-Score. Similarly, the End Localization Score (EL-Score) is defined based on the relative distance between the predicted and the ground-truth end time.

TABLE IV
RESULTS OF USING SKELETON AND 3D POSE ON THE OAD DATASET

Input	$F1$-	SL-	EL-
RGB	0.578	0.341	0.371
3D Pose	0.598	0.343	0.372
Skeleton	**0.694**	**0.475**	**0.495**

TABLE V
RESULTS OF COMBINATIONS WITH DIFFERENT MODALITIES ON THE OAD DATASET

Modality\Scores	$F1$-	SL-	EL-
RGB	0.578	0.341	0.371
Optical Flow	0.704	0.472	0.489
Skeleton	0.694	0.475	0.495
RGB + Optical Flow	0.741	0.491	0.517
RGB + Skeleton	0.692	0.442	0.470
Optical Flow + Skeleton	**0.795**	**0.576**	**0.597**
RGB + Optical Flow + Skeleton	0.785	0.561	0.577

2) Comparison of Skeleton and Estimated 3D Pose: Compared with 2D RGB image, one of the advantages of skeleton data is that it contains 3D depth information. Recent 3D pose estimation works [83], [84] provide another option to extract 3D pose from RGB data. Here we compare the performance of action detection using skeleton data collected from Kinect and 3D pose derived from RGB data. We first adopt the state-of-the-art 3D pose estimation method [83] to extract 3D pose for OAD dataset. Then, we train the proposed models on two different inputs using the same network and settings. We also compare the results with directly using the RGB data. Table IV shows the results of three input data. We can see that the results of using 3D pose is slightly better than using RGB data. It indicates that although 3D pose is derived from RGB data but its high-level motion representation benefits action detection task. However, compared with the accurate skeleton which obtained from Kinect using additional depth information, the results of using 3D pose are more inferior than using skeleton data.

3) Combination of Different Modalities: Since our model supports different kinds of modalities as the input, we investigate the performance of these modalities and their combination on the OAD dataset. In the testing phrase, we fuse the results from different modality branches by averaging the scores. From Table V it is found that: (i) For single modality model, Optical Flow and Skeleton networks achieve similar performance (0.704 vs. 0.694 $F1$-Score) on the OAD dataset while the results of RGB network (0.578 $F1$-Score) are relatively worse. Such a result demonstrates that motion information is the key factor for human action detection. (ii) Most modality combinations boost the detection performance over the single modality model. For example, combining RGB and optical flow achieves 0.741 $F1$-Score and fusion of optical flow and skeleton improves the performance to 0.795. It reflects that different modalities provide complementary information for the action detection task. (iii) Combining all of three modalities leads to the $F1$-Score of 0.785, which is slightly lower than the result of combining optical flow and skeleton. This result demonstrates that, the combination of optical flow and skeleton brings about more compact and effective features. Thus, in the following experiments, we select the results of combining optical flow and skeleton for multiple modalities model.

TABLE VI
RESULTS WITH AND WITHOUT MOTION INSPECTING (MI) LAYER

Methods\Scores	$F1$-	SL-	EL-
Skeleton w/o MI	0.653	0.418	0.443
Skeleton with MI	0.694	0.475	0.495
Optical Flow + Skeleton w/o MI	0.769	0.540	0.570
Optical Flow + Skeleton with MI	**0.795**	**0.576**	**0.597**

TABLE VII
RESULTS WITH AND WITHOUT SOFT SELECTOR (SS) LAYER

Methods\Scores	$F1$-	SL-	EL-
Skeleton w/o SS	0.621	0.389	0.412
Skeleton with SS	0.694	0.475	0.495
Optical Flow + Skeleton w/o SS	0.761	0.530	0.560
Optical Flow + Skeleton with SS	**0.795**	**0.576**	**0.597**

TABLE VIII
RESULTS WITH AND WITHOUT LSTM SUBNETWORK

Methods\Scores	$F1$-	SL-	EL-
Skeleton w/o LSTM	0.435	0.233	0.228
Skeleton with LSTM	0.694	0.475	0.495
Optical Flow w/o LSTM	0.569	0.287	0.325
Optical Flow with LSTM	0.704	0.472	0.489
Optical Flow + Skeleton w/o LSTM	0.617	0.342	0.370
Optical Flow + Skeleton with LSTM	**0.795**	**0.576**	**0.597**

4) Effectiveness of Motion Inspecting Layer and Soft Selector Layer: As the Motion Inspecting (MI) and Soft Selector (SS) layers play an important role in our model, we perform ablation analysis to verify their effectiveness, respectively. To evaluate the influence of the MI layer, we implement a simplified version of our method by removing the MI layer of the skeleton network and directly feed the raw skeleton data into the LSTM layers. Table VI illustrates the $F1$-, SL- and EL-Scores of models with and without the MI layer. It is observed that the incorporation of the MI layer brings about significant improvement.

Analogously, we implement a model removing the Soft Selector (SS) module in our networks to verify the functionality of the SS layer and directly linking FC2 and FC3 layers. Table VII compares our method with and without SS layer on the OAD Dataset. Note that we use the same parameters for two settings. The results show that the SS layer consistently improves the final results for both skeleton and multi-modality models.

5) Effectiveness of LSTM Subnetwork: The stacked LSTM subnetwork is responsible for extracting temporal dynamics based on the previous representative features from the convolutional network or Motional Inspecting layer. To validate

TABLE IX
$F1$-SCORE FOR INDIVIDUAL ACTIONS ON THE OAD DATASET

Actions	SVM-SW	RNN-SW [20]	JCR-RNN [30]	RF+ST [32]	SM-MT RNN	MM-MT RNN
drinking	0.146	0.441	0.574	0.517	**0.761**	0.538
eating	0.465	0.550	0.523	0.645	0.625	**0.658**
writing	0.645	0.859	0.822	0.803	0.823	**0.892**
opening cupboard	0.308	0.321	0.495	0.555	0.575	**0.643**
washing hands	0.562	0.668	0.718	**0.860**	0.705	0.800
opening microwave	0.607	0.665	0.703	0.610	0.655	**0.780**
sweeping	0.461	0.590	0.643	0.437	0.576	**0.882**
gargling	0.437	0.550	0.623	**0.722**	0.598	0.658
throwing trash	0.554	0.674	0.459	0.688	0.516	**0.748**
wiping	0.857	0.747	0.780	0.977	0.771	**0.943**
Overall	0.540	0.600	0.653	0.672	0.694	**0.795**

TABLE X
RESULTS OF DIFFERENT METHODS ON THE OAD DATASET

Methods\Scores	$F1$-	SL-	EL-
SVM-SW	0.540	0.316	0.325
RNN-SW [20]	0.600	0.366	0.376
JCR-RNN [30]	0.653	0.418	0.443
SM-MT RNN (skeleton)	0.694	0.475	0.495
RF+ST [32]	0.672	0.445	0.432
MM-ST RNN (skeleton + optical flow)	0.748	0.515	0.522
MM-MT RNN (skeleton + optical flow)	**0.795**	**0.576**	**0.597**

the effectiveness of LSTM module in temporal modeling, we also conduct an ablation experiment to compare with a framewise method by replacing LSTM layers with fully-connected layers. The results are shown in Table VIII. We can see that the proposed method achieves better results for all skeleton, flow and multi-modality models. This demonstrates that LSTM module could capture effective temporal dynamics which benefits the final action recognition and detection.

6) Detection Performance Comparisons: Table X shows the $F1$-, SL- and EL-Scores of our method and other compared methods on the OAD dataset. Methods in the top part of the Table X use only skeleton data while methods in the bottom use both skeleton and RGB data. From Table X, we observe: (i) Our single modality model SM-MT RNN achieves the best performance among all the models using only skeleton data. It is demonstrated that our RNN-based architecture is more efficient than the classical sliding window scheme, and the Motion Inspecting Layer successfully captures the dynamics of skeleton data. (ii) Our multi-modality models improve the detection performance over all single modality models and achieve better results than the random forest method RF+ST [32] in the same setting. (iii) The MM-MT RNN further improves over the MM-ST RNN (Multi-Modality Single-Task RNN), which only aims at classification without considering regression. It reveals that incorporating the regression task into the network and jointly optimizing classification-regression both boost the localization and the detection accuracy.

We also show the results of each action class and the average $F1$-Score of all actions on the OAD Dataset in Table IX. It is seen that, the proposed scheme achieves the best detection accuracy in most action classes and gets the highest overall score.

For the G3D dataset, we evaluate the performance in terms of three types of scores for seven categories of sequences with different modalities in Table XI. Note that due to the low quality of the RGB data in G3D dataset (many missing RGB frames), the results of RGB and Optical Flow is much worse than the results of skeleton in most cases. This also affects the final combination results.

In the Table XII and XIII, we compare our methods using only skeleton data with other methods. The results are consistent with those on the OAD dataset. We also compare these methods in Table XIV using the evaluation metric action-based $F1$ as defined in [76], which treats the detection of an action as correct when the predicted start point is within 4 frames of the ground-truth start point for that action. Note that the action-based $F1$ only considers the accuracy of the start point. The method in [76] uses a traditional boosting algorithm [85] and its scores are significantly lower than other methods.

D. Evaluation in Action Forecast

In this section, we show the evaluation results of the proposed methods on action forecast task. We first introduce the evaluation criteria and compared forecasting baseline methods. Then we illustrate the forecast performance of different methods.

1) Evaluation Criteria: As explained in Section III, the system is expected to forecast whether the action will start or end within T frames prior to its occurrence. To be considered as a true positive start forecast, the forecast should not only predict the impending action class, but also do that within a reasonable interval, *i.e.*, $[t_{start}-T, t_{start}]$ for an action starting at t_{start}. This rule is also applied to the end forecast. We use the Precision-Recall Curve to evaluate the performance of action forecast methods. Note that both precision and recall are calculated on the frame-level for all frames.

2) Compared Methods: We use a simple strategy [30] to forecast for some detection baseline methods. For SVM-SW, RNN-SW, they output the probability $q_{t,j}$ for each action

TABLE XI
$F1$-SCORE ON THE G3D DATASET WITH DIFFERENT MODALITIES

Modality\Action Category	Fighting	Golf	Tennis	Bowling	FPS	Driving	Misc
RGB	0.489	1.000	0.560	0.703	0.589	1.000	0.910
Optical Flow	0.614	1.000	0.629	0.889	**0.903**	1.000	0.951
Skeleton	**0.860**	1.000	**0.829**	**1.000**	0.627	1.000	**0.986**
Optical Flow + Skeleton	0.639	1.000	0.772	0.667	0.703	1.000	0.967
RGB + Optical Flow + Skeleton	0.695	1.000	0.785	0.629	0.635	1.000	0.952

TABLE XII
$F1$-SCORE ON THE G3D DATASET

Action Category	SVM -SW	RNN -SW [20]	JCR RNN [30]	SM-MT RNN
Fighting	0.486	0.613	0.735	**0.860**
Golf	0.680	0.745	0.967	**1.000**
Tennis	0.598	0.480	0.788	**0.829**
Bowling	0.667	0.889	**1.000**	**1.000**
FPS	0.571	0.581	0.523	**0.627**
Driving	1.000	1.000	1.000	1.000
Misc	0.712	0.742	0.862	**0.986**

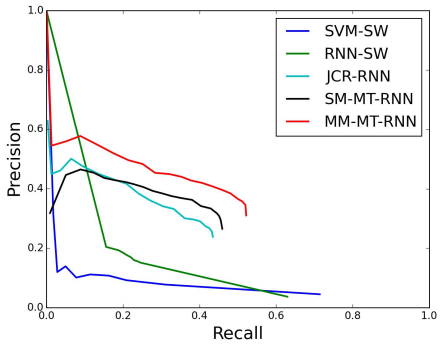
TABLE XIII
SL- AND EL-SCORES ON THE G3D DATASET

Action Category	Scores	SVM -SW	RNN -SW [20]	JCR RNN [30]	SM-MT RNN
Fighting	SL-	0.318	0.412	0.528	**0.635**
	EL-	0.328	0.419	0.557	**0.723**
Golf	SL-	0.553	0.635	0.793	**0.846**
	EL-	0.524	0.656	0.836	**0.867**
Tennis	SL-	0.444	0.338	0.665	**0.666**
	EL-	0.460	0.333	0.667	**0.675**
Bowling	SL-	0.612	0.777	**0.959**	0.869
	EL-	0.550	0.713	**0.861**	0.809
FPS	SL-	0.351	0.388	0.311	**0.402**
	EL-	0.353	0.393	0.327	**0.431**
Driving	SL-	**0.991**	0.983	0.955	0.929
	EL-	**0.975**	**0.975**	**0.975**	0.954
Misc	SL-	0.487	0.593	0.614	**0.815**
	EL-	0.515	0.612	0.766	**0.928**

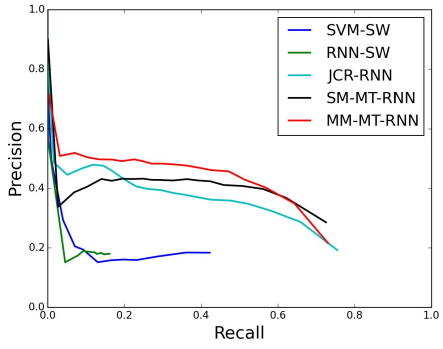
TABLE XIV
ACTION-BASED $F1$ [76] ON THE G3D DATASET

Action Category	G3D [77]	SVM -SW	RNN -SW [20]	JCR RNN [30]	SM-MT RNN
Fighting	58.54	76.72	83.28	**96.18**	96.00
Golf	11.88	45.00	55.00	70.00	**76.67**
Tennis	14.85	37.57	36.68	62.38	**70.48**
Bowling	31.58	22.22	44.44	**66.67**	44.44
FPS	13.65	35.35	39.89	33.85	**43.37**
Driving	2.50	39.99	**50.00**	**50.00**	**50.00**
Misc	18.13	53.32	65.24	**86.19**	81.90

class j at each time step t. At time t, when the probability $q_{t,j}$ of action class j is larger than a predefined threshold β_s, we consider that the action of class j will start soon.



(a)



(b)

Fig. 7. The Precision-Recall curves of the start and end time forecast by different methods on the OAD dataset. Overall MM-MT RNN achieves the best performance. This figure is best viewed in color. (a) Forecast of start. (b) Forecast of end.

Similarly, during an ongoing period of the action of class j, when the probability $q_{t,j}$ is smaller than another threshold β_e, we consider this action to end soon.

3) Forecast Performance: The peak point of the regressed confidence curve is considered as the start/end point in the test. We set a confidence threshold θ_s (or θ_e) according to the sensitivity requirement of the system to predict the start (or end) point. When the current confidence value is higher than θ_s but ahead of the peak, this frame forecasts the start of an action. By adjusting the confidence thresholds θ_s and θ_e, we draw the Precision-Recall curves for our method. Similarly, we define the confidence thresholds β_s and β_e and draw the curves for the detection baselines by adjusting β_s and β_e. As shown in Fig. 7, the performance of the detection baselines (SVM-SW and RNN-SW) is significantly inferior to our methods. The result suggests that only using the traditional detection probability is not enough for forecasting. One important reason is that the frames before the start time

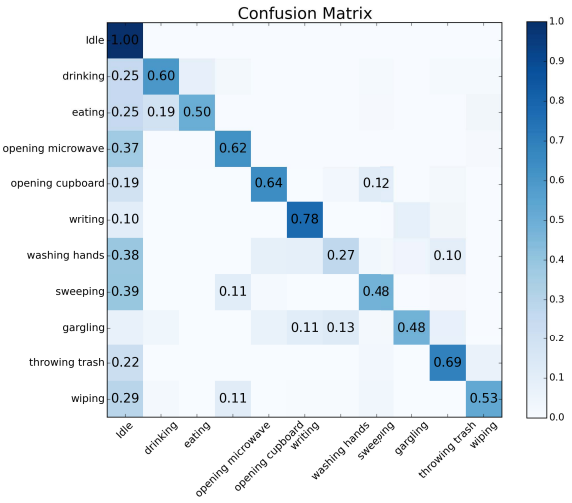


Fig. 8. Confusion Matrix of start forecast on the OAD dataset. Vertical axis: ground-truth class; Horizontal axis: predicted class.

are simply treated as background samples in the baselines but actually they contain evidences. While in our regression task, we deal with these frames using different confidence values to guide the network to explore the hidden starting or ending patterns of actions. In addition, our method also achieves better performance over JCR-RNN, which demonstrates the effectiveness of our multiple modalities fusion and the design of our network architecture. We also note that the forecast precision of all the methods is not very high even though our method is much better. It is because the forecast problem itself is very difficult. For example, when a person is writing on the board, it is difficult to forecast whether he will finish writing soon.

Fig. 8 shows the confusion matrix of the start forecast by our proposed method. This confusion matrix represents the relationships between the predicted start action class and the ground-truth class. The shown confusion matrix is obtained by averaging among all test videos when the recall rate equals to 40%. Although there are some missed or wrong forecasts, most of the forecasts are correct. In addition, there are a few interesting observations. (i) The action *eating* and *drinking* may have similar poses before they start. (ii) Action *gargling* and *washing hands* are also easy to be mixed up when forecasting, since these two actions both need to turn on the tap before starting.

E. Comparison of Running Speeds

We compare the running speeds of different methods. Table XV shows the average running time on nine long sequences, which has 3200 frames on average. For a fair comparison, all the methods are running on the CPU (except 7.41s in the brackets for MM-MT RNN is measured on a single TitanX GPU). SVM-SW is the fastest because of its compact model compared with the deep learning methods. The RNN-SW runs slower than our methods due to its sliding window design. It is noticed that the running speed for the action detection based on only skeleton input is rather fast, being 1145fps for the SM-MT RNN approach. It is because

TABLE XV
AVERAGE RUNNING TIME (SECONDS PER SEQUENCE)

SVM-SW	RNN-SW [20]	JCR-RNN [30]	SM-MT RNN	MM-MT RNN
1.05	3.14	2.60	2.79	1403.95 (7.41)

of the low dimension of skeleton ($25 \times 3 = 75$ for each frame) compared with RGB input. It makes the skeleton based online action detection much more attractive for real-time applications. At the same time, although incorporating RGB data (MM-MT RNN) increases the computational complexity due to the high dimension of RGB frames, it improves the detection and forecasting performance a lot. Note that the running time can also be reduced a lot for MM-MT RNN with a GPU. Thus, our method provides users an option to make a trade-off between whether to combine RGB data or not.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a Multi-Modality Multi-Task Recurrent Neural Network to recognize the action type and better localize the start and end points on the fly. We first design different temporal modeling networks for different modalities, *e.g.*, a deep Convolutional network for RGB data and a Motion Inspecting network for skeleton data. Then we leverage the merits of the deep LSTM network to capture the complex long-range temporal dynamics without the conventional sliding window design. The multi-task objective function helps classify and localize the start and end time of actions more accurately. At the same time, incorporating the regression task network, our joint classification-regression model is capable of forecasting the occurrence of actions in advance. Experiments on two datasets demonstrate the effectiveness of the proposed method.

Although fusing different modalities in an end-to-end deep network to capture temporal dynamics of actions, our framework does not pay attention to describing human-object interactions explicitly, which may lead to missing the great potential to model and differentiate some similar actions from microscopic side. Therefore, one promising future direction is to extract human-object interaction information features into our model to further improve the detection and forecast performance.

VII. ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NVIDIA Corporation with the GPU for this research. The key technology was done at Microsoft Research Asia.

REFERENCES

[1] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Understand.*, vol. 115, no. 2, pp. 224–241, Feb. 2011.
[2] D. Oneata, J. Verbeek, and C. Schmid, "The LEAR submission at THUMOS 2014," in *Proc. THUMOS Action Recognit. Challenge*, 2014.
[3] G. Yu, J. Yuan, and Z. Liu, "Propagative Hough voting for human activity detection and recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 87–98, Jan. 2015.
[4] D. P. Barrett and J. M. Siskind, "Action recognition by time series of retinotopic appearance and motion features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2250–2263, Dec. 2016.

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

[6] M. Hoai and F. De la Torre, "Max-margin early event detectors," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 191–202, 2014.

[7] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," in *Proc. THUMOS Action Recognit. Challenge*, 2014.

[8] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2642–2649.

[9] D. Oneata, J. Verbeek, and C. Schmid, "Efficient action localization with approximately normalized Fisher vectors," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2545–2552.

[10] K. Soomro, H. Idrees, and M. Shah, "Predicting the where and what of actors and actions through online action localization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2648–2657.

[11] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 461–470.

[12] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2625–2634.

[13] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2678–2687.

[14] 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 *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, June 2016, pp. 1961–1970.

[15] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in LSTMs for activity detection and early detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1942–1950.

[16] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[17] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1297–1304.

[18] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 370–385.

[19] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1110–1118.

[20] W. Zhu *et al.*, "Co-Occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI*, 2016, pp. 3697–3703.

[21] A. Sharaf, M. Torki, M. E. Hussein, and M. El-Saban, "Real-time multi-scale action detection from 3D skeleton data," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 998–1005.

[22] G. Garcia-Hernando and T.-K. Kim. (2017). "Transition forests: Learning discriminative temporal transitions for action recognition and detection." [Online]. Available: https://arxiv.org/abs/1607.02737

[23] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2752–2759.

[24] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[25] R. Vemulapalli and R. Chellappa, "Rolling rotations for recognizing human actions from 3D skeletal data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4471–4479.

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1290–1297.

[27] X. Cai, W. Zhou, L. Wu, J. Luo, and H. Li, "Effective active skeleton representation for low latency human action recognition," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 141–154, Feb. 2016.

[28] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.

[29] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. AAAI*, 2017, pp. 4263–4270.

[30] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 203–220.

[31] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3054–3062.

[32] S. Baek, K. I. Kim, and T.-K. Kim, "Real-time online action detection forests using spatio-temporal contexts," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2017, pp. 158–167.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1010–1019.

[35] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[36] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.

[37] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[38] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[39] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[40] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1933–1941.

[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, 2017, pp. 4278–4284.

[42] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[44] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "VLAD3: Encoding dynamics of deep features for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1951–1960.

[45] B. Fernando, P. Anderson, M. Hutter, and S. Gould, "Discriminative hierarchical rank pooling for activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1924–1932.

[46] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3034–3042.

[47] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[48] P. Siva and T. Xiang, "Weakly supervised action detection," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.

[49] L. Wang, Z. Wang, Y. Xiong, and Y. Qiao, "CUHK&SIAT submission for thumos15 action recognition challenge," in *Proc. THUMOS Action Recognit. Challenge*, 2015.

[50] M. Jain, J. van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 740–747.

[51] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1302–1311.

[52] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "DAPs: Deep action proposals for action understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 768–784.

[53] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu, "Concurrent action detection with structural prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3136–3143.

[54] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1036–1043.

[55] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient nonlinear Markov models for human motion," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1314–1321.

[56] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 724–731.
[57] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2466–2472.
[58] Y. Goutsu, W. Takano, and Y. Nakamura, "Motion recognition employing multiple Kernel learning of Fisher vectors using local skeleton features," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 321–328.
[59] L. Tao and R. Vidal, "Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 303–311.
[60] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang, "Multimodal multipart learning for action recognition in depth videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2123–2129, Oct. 2016.
[61] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 816–833.
[62] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. IEEE Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 579–583.
[63] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 624–628, May 2017.
[64] M. Meshry, M. E. Hussein, and M. Torki, "Linear-time online action detection from 3D skeletal data using bags of gesturelets," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2016, pp. 1–9.
[65] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, p. 1.
[66] Q. Wu, Z. Wang, F. Deng, Z. Chi, and D. D. Feng, "Realistic human action recognition with multimodal feature selection and fusion," *IEEE Trans. Syst., Man, Cybernetics, Syst.*, vol. 43, no. 4, pp. 875–885, Jul. 2013.
[67] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen, "Two streams recurrent neural networks for large-scale continuous gesture recognition," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 31–36.
[68] G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-D convolution and convolutional LSTM," *IEEE Access*, vol. 5, pp. 4517–4524, 2017.
[69] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 5–13.
[70] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*. Hoboken, NJ, USA: Wiley, 2001.
[71] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–9.
[73] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[74] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi, "Joint classification-regression forests for spatially structured multi-object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 870–881.
[75] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Accurate object detection with joint classification-regression random forests," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 923–930.
[76] V. Bloom, D. Makris, and V. Argyriou, "G3D: A gaming action dataset and real time action recognition evaluation framework," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 7–12.
[77] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2010, pp. 9–14.
[78] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Proc. IEEE Int. Conf. Computer Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 28–35.
[79] Y.-G. Jiang *et al.* (2014). *THUMOS Challenge: Action Recognition With a Large Number of Classes*. [Online]. Available: http://crcv.ucf.edu/THUMOS14/
[80] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 961–970.
[81] K. Soomro, A. R. Zamir, and M. Shah. (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild." [Online]. Available: https://arxiv.org/abs/1212.0402
[82] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[83] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1–10.
[84] D. Mehta *et al.*, "VNect: Real-time 3D human pose estimation with a single RGB camera," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 44. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/VNect/
[85] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 148–156.

Jiaying Liu (S'08–M'10–SM'17) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, in 2005 and the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2010.

She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. She has authored over 100 technical articles in refereed journals and proceedings, and holds 20 granted patents. Her current research interests include image/video processing, compression, and computer vision.

She was a Visiting Scholar with University of Southern California, Los Angeles, from 2007 to 2008. She was a Visiting Researcher with Microsoft Research Asia in 2015, supported by Star Track for Young Faculties. She has also served as a TC Member in the IEEE CAS MSA and APSIPA IVM, and an APSIPA Distinguished Lecturer from 2016 to 2017. She is a CCF Senior Member.

Yanghao Li (S'17) received the B.S. degree in computer science from Peking University, Beijing, China, in 2015, where he is currently pursuing the master's degree with the Institute of Computer Science and Technology.

His current research interests include action analysis and computer vision.

Sijie Song (S'17) received the B.S. degree in computer science from Peking University, Beijing, China, in 2016, where she is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology. Her research interests include computer vision and image processing.

Junliang Xing (M'09) received dual B.S. degrees in computer science and mathematics from Xi'an Jiaotong University, Shaanxi, China, in 2007, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2012. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision problems related to faces and humans.

Cuiling Lan received the B.Sc. degree in electrical engineering and the Ph.D. degree in intelligent information processing from Xidian University, China, in 2008 and 2014, respectively. She joined Microsoft Research Asia in 2014. Her research interests include computer vision, image and video compression, and transmission.

Wenjun (Kevin) Zeng (M'97–SM'03–F'12) received the B.E. degree from Tsinghua University, the M.S. degree from University of Notre Dame, and the Ph.D. degree from Princeton University. He has been leading the video analytics research empowering the Microsoft Cognitive Services and Azure Media Analytics Services since 2014. He had been with PacketVideo Corporation, Sharp Labs of America, Bell Labs, and Panasonic Technology. He was with University of Missouri from 2003 to 2016, most recently as a Full Professor. He has contributed significantly to the development of international standards (ISO MPEG, JPEG2000, and OMA). He is currently a Principal Research Manager and a Member of the Senior Leadership Team, Microsoft Research Asia. His current research interest includes mobile-cloud media computing, computer vision, social network/media analysis, and multimedia communications and security.

He was a recipient of several best paper awards (e.g., the IEEE VCIP'2016, the IEEE ComSoC MMTC 2015 Best Journal Paper, and ACM ICMCS'2012). He served as the Steering Committee Chair of IEEE ICME in 2010 and 2011, and is serving or has served as the General Chair or the TPC Chair for several IEEE conferences (e.g., ICME'2018 and ICIP'2017). He is an Associate Editor-in-Chief of *IEEE Multimedia Magazine*, and was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON MULTIMEDIA. He was a Special Issue Guest Editor for PROCEEDINGS OF THE IEEE, TMM, ACM TOMCCAP, TCSVT, and *IEEE Communications Magazine*. He was on the Steering Committee of IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TMM.