SSNet: Scale Selection Network for Online 3D Action Prediction

Jun Liu[†], Amir Shahroudy[†], Gang Wang[‡], Ling-Yu Duan[§], Alex C. Kot[†]
[†] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[‡] Alibaba Group, Hangzhou, China
[§] National Engineering Lab for Video Technology, Peking University, Beijing, China
{jliu029, amir3, wanggang, eackot}@ntu.edu.sg, lingyu@pku.edu.cn

Abstract

In action prediction (early action recognition), the goal is to predict the class label of an ongoing action using its observed part so far. In this paper, we focus on online action prediction in streaming 3D skeleton sequences. A dilated convolutional network is introduced to model the motion dynamics in temporal dimension via a sliding window over the time axis. As there are significant temporal scale variations of the observed part of the ongoing action at different progress levels, we propose a novel window scale selection scheme to make our network focus on the performed part of the ongoing action and try to suppress the noise from the previous actions at each time step. Furthermore, an activation sharing scheme is proposed to deal with the overlapping computations among the adjacent steps, which allows our model to run more efficiently. The extensive experiments on two challenging datasets show the effectiveness of the proposed action prediction framework.

1. Introduction

Action prediction is to recognize the class label of an ongoing activity when only a part of it is perceived. Predicting actions before they get completely performed is a subset of a broader research domain on human activity analysis. It has attracted a lot of attention due to its wide range of applications in security surveillance, human-machine interaction, patient monitoring, etc [23, 4].

Most of the existing works in literature [23, 27, 32] focus on action prediction in the well-segmented videos, for which each video contains only one action instance. However, in practical scenarios, such as online human-machine interaction systems, plenty of action instances are contained in a streaming sequence, which are not segmented. In this paper, we address this more challenging task: "online action prediction in untrimmed video", i.e. we want to recognize the current ongoing action from the observed part of it at each temporal step of the data streaming, which can include multiple actions, as shown in Figure 1(a).

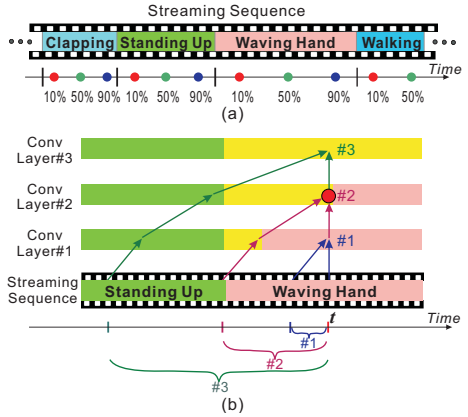


Figure 1. Fig (a) shows an untrimmed streaming sequence containing multiple action instances. We need to recognize the current ongoing action at each time step when only a part of it (eg. 10%) is performed. Fig (b) depicts our SSNet for online 3D action prediction. At time t, only a part of the action *waving hand* is observed. Our SSNet selects the convolutional layer #2 rather than #3 for prediction, as the perception window of #2 mainly covers the performed part of current action, while #3 involves too many frames from the previous action which can interfere the prediction at t.

Biological studies [25] show that the skeleton data is informative enough for representing human behaviors, even without appearance information [80]. Human activities are naturally performed in 3D space, thus 3D skeleton data is suitable for representing human actions [46]. These 3D skeleton information can be easily and effectively acquired in real-time with the low-cost depth sensor, such as Kinect [59, 78]. As a result, activity analysis with 3D skeleton data becomes popular [19, 49, 62, 41, 81, 57, 43] due to its succinctness, high level representation, and robustness against variations in viewpoints, clothing textures, illumination, and background [11, 42, 23].

In this paper, we investigate real-time action prediction

Corresponding authors: Jun Liu and Ling-Yu Duan.

with the continuous 3D skeleton data. To predict the class label of the current ongoing action at each temporal step, we adopt a sliding window method over the frames of the streaming skeleton sequence, and the data frames inside the window are used to perform action prediction at each step.

Sliding window based design has been widely used in a series of computer vision tasks, such as object recognition [47], pedestrian detection [12], activity detection [48, 56, 77, 22], etc. Most of these works adopt one fixed scale or combine multi-scale multi-pass scans at each sliding position. However, in our online action prediction task, we need to predict the ongoing action at each observation ratio, while there are significant temporal scale variations in the observed part of the ongoing action at different progress levels. This makes it quite difficult to determine the scale of the sliding window. In addition, the untrimmed streaming sequence may contain multiple action instances (see Figure 1(a)). The order of the actions can be arbitrary, and the duration of different instances are often not the same. Moreover, the observed (per whole) ratio of the ongoing action changes over time, which makes it even more challenging to obtain a proper temporal window scale for online prediction. For example, at an early temporal stage, it is beneficial to use a relatively small scale, because the larger window sizes may include frames from the previous action instance which can mislead the recognition of the current instance. Conversely, if a large part of the current action has been already observed, it is beneficial to use a larger window size to cover more of its performed parts in order to achieve a reliable prediction.

To deal with the above-mentioned challenges, we propose a novel Scale Selection Network (SSNet) in this paper for online action prediction. Rather than using a fixed scale or multi-scale multi-pass scans at each time step, we supervise our network to choose the proper window scale dynamically at each step to cover the performed part of the current action instance. In our method, the network predicts the ongoing action at each frame. Beside predicting the class label, it also regresses the temporal distance to the beginning of current action instance, which indicates the performed part of the ongoing action. Thus, at the next frame, we can utilize it as the temporal window scale for class prediction.

Specifically, we apply convolutional analysis in temporal dimension to model the motion dynamics over the frames for 3D action prediction. A hierarchical architecture with dilated convolution layers is leveraged to learn a comprehensive representation over the frames within each perception window, such that different layers in our SSNet correspond to different temporal scales, as shown in Figure 1(b). Therefore, at each time step, our network selects the *proper* convolutional layer which covers the most similar window scale regressed by its previous step. Then the activations of this layer can be used for action prediction. The proposed SSNet is designed to select the proper window in order to cover the performed part of current action and tries to suppress the noisy data from the previous ones, hence it can produce reliable predictions at each step. To the best of our knowledge, this is the first convolutional model with explicit temporal scale selection as its fundamental capability for handling scale variations in online activity analysis.

In many existing methods which use sliding windows, the computational efficiencies are relatively low due to the overlapping design and exhaustive multi-scale multi-round scan. In our method, the action prediction is performed with a regressed scale at each step, which avoids multi-pass scan. So the action prediction and scale selection are performed by a single convolutional network very efficiently. Moreover, we introduce an activation sharing scheme to deal with the overlapping computations over different time steps, which makes our SSNet run very fast for real-time online prediction.

We summarize the main contributions as: (1) We study the new problem of real-time online action prediction in continuous 3D skeleton streams by leveraging convolutional analysis in temporal dimension. (2) The proposed Scale Selection Network is capable of dealing with the scale variations of the observed portion of the ongoing action at different time steps. We propose a scale selection scheme to let our network choose the proper temporal scale at each step, such that the network can mainly focus on the performed part of current action, and try to avoid the noise from the previous action samples. (3) Our framework is very efficient for online action analysis due to the computation sharing over different time steps. (4) We perform action prediction with our SSNet which is end-to-end trainable, rather than using expensive multi-stage or multi-network design at each step. (5) Our method achieves superior performance on two challenging datasets for 3D activity analysis.

2. Related Work

3D Action Recognition. After the development of cheap and easy-to-use depth sensors, such as Kinect and XTion, human action recognition in 3D skeleton sequences becomes very popular [1, 79], and a series of hand-crafted features [72, 63, 75, 64, 52, 13, 70, 67] and deep learning based approaches [11, 83, 51, 30, 42, 26] have been proposed. Most of the existing 3D action recognition methods [63, 51, 57, 69, 24, 66, 10, 29] take fully observed segmented videos as input (each sample contains one full action instance), and output a class label. The proposed online 3D action prediction method takes one step forward in dealing with numerous action instances occurring in the untrimmed sequences, for which the current ongoing action can be partly observed.

There are very limited number of 3D action recognition

methods [3] which attempted untrimmed sequences. Different from these works, the proposed SSNet framework predicts the class of the current ongoing action by utilizing its predicted observation ratio.

Action Prediction. Recognizing (predicting) an action before it is fully performed has attracted a lot of attention recently [31, 50, 4, 71, 36, 27, 28]. Ryoo *et al*. [50] represented each action as an integral histogram of spatio-temporal features for activity prediction. Li *et al*. [37] designed a predictive accumulative function. Ke *et al*. [27] extracted deep features in optical flow images for activity prediction.

Recently, Hu *et al*. [23] explored to incorporate 3D skeleton information for real-time action prediction in *well-segmented* sequences, i.e., each sequence includes only one action. A soft regression strategy was introduced in their work for action prediction. However, their approach was not suitable for online 3D action prediction in *untrimmed* continuous sequence, which contains multiple action instances.

Action Analysis in Untrimmed Videos. Beside the online action prediction task, the problem of temporal action detection [68, 73, 48, 33, 7, 53, 68, 14, 54, 16, 15, 65, 82] also copes with untrimmed videos. Several methods attempted online detection [38], while most action detection approaches are developed for handling offline mode which conducts detection after observing the whole long sequence [48, 35, 56]. Our task is different from action detection, as action detection mainly addresses accurate spatio-temporal segmentation, while action prediction focuses more on predicting the class of the current ongoing action timely from its observed part, even when only a small ratio of it is performed.

The sliding window based design [77, 53, 2, 22] and action proposals [76] have been adopted for action detection. Zanfir *et al*. [77] used a sliding window with one fixed scale (obtained by cross validation) for action detection. Shou *et al*. [55] adopted multi-scale windows for action detection via multi-stage networks. Differently, in our action prediction task, determining the scale of the window is challenging due to the scale variations of the observed part of the ongoing action. Also, instead of using one fixed scale [77] or multi-scale multi-pass scans [55, 84], we propose a novel SSNet for online prediction, which is supervised to choose the proper window for prediction at each temporal step. Moreover, the redundant computations are efficiently shared over different steps in our method.

3. SSNet: Scale Selection Network

In this section, we introduce the proposed network architecture, Scale Selection Network (SSNet), for online 3D action prediction. The overall schema of this method is illustrated in Figure 2. In the proposed network, a hierarchy of one dimensional convolutions are performed in temporal domain to model the motion dynamics over the frames. Inputs of SSNet are the frames within a time window at each time step. In order to tackle the scale variation in the partially observed action at different time steps, a scale selection method is proposed, which enables our SSNet to focus on the observed part of the ongoing action by picking the most suitable convolutional layers.

3.1. Temporal Modeling with Convolutional Layers

Recently, convolutional networks [34, 9, 18] have proven their superior strength in modeling the time series data [60, 8, 61]. For example, van den Oord *et al*. [60] proposed a convolutional model, called WaveNet, for audio signal generation, and Dauphin *et al*. [8] introduced a convolutional network for time series in language sequential modeling. Inspired by the success of convolutional approaches in the analysis of temporal sequential data, we leverage a stack of 1-D convolutional layers to model the motion dynamics and context dependencies over the video sequence frames, and inspired by the WaveNet model, we propose a network for the 3D action prediction task. Specifically, previous works [58, 5] suggest that there are often hierarchical structures in the motion patterns over temporal axis, which play an important role in action analysis, hence, we design our convolutional model with a hierarchical structure.

In our method, the convolutional model is used to learn a comprehensive representation over all the frames within a temporal window, and then this representation can be used for action prediction. The main building blocks of our model are dilated causal convolutions. Causal design [60] indicates action prediction at time t is based on the available information before t (including t) without using future information. Dilated convolution [74] means the convolutional filter works over a larger field than the filter's length, and some input values inside the field are skipped with a certain step.

This is intuitive for action analysis, since the running time for longer actions can be very long and the convolutional network needs to be able to cover a large receptive field. Applying standard convolution, the network needs more layers or larger filter sizes to achieve a broader receptive field. However, both of these significantly increase the number of model parameters. Dilated convolution, supports expansion of the receptive field very efficiently, without bringing more parameters [74]. In addition, it does not need any extra pooling operations, thus it can well maintain the ordering information of the inputs [74]. Therefore, dilated convolution is suitable for the task of action prediction.

The proposed model is depicted in Figure 2. Each convolution operation is performed over two input nodes with the dilation degree set to d, where $d = 1$ represents the standard convolution. The dilation degree increases exponentially over the layers in the network, i.e., we set d to

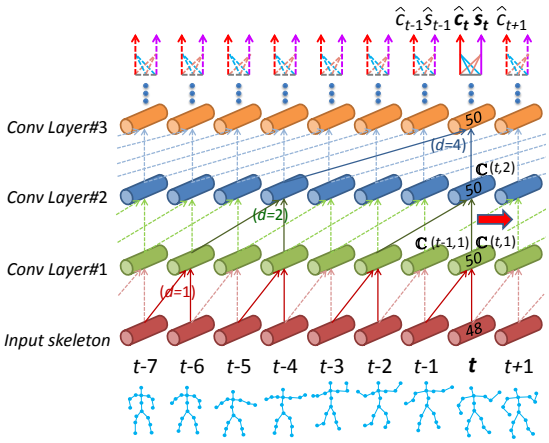


Figure 2. Schema of the proposed SSNet for action prediction over the temporal axis. Solid lines denote the SSNet links activated at current step t, and dashed lines indicate the links activated at other time steps. Here we only show 3 convolutional layers for clarity. At each step, SSNet predicts the class ($\hat{c}_t$) of the ongoing action, and also estimates the distance ($\hat{s}_t$) to current action's start point. Calculation details of $\hat{c}_t$ and $\hat{s}_t$ are shown in Figure 3. Convolutional filters are shared at each layer, yet different across layers.

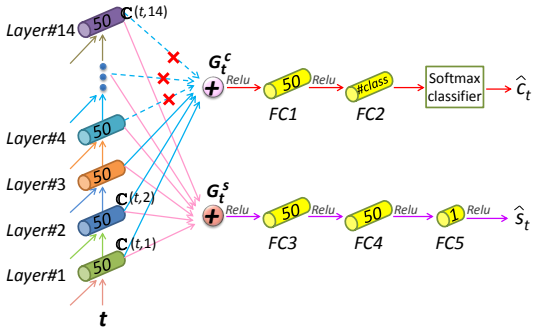


Figure 3. Details of our SSNet which jointly predicts the class label $\hat{c}_t$ and regresses the start point's distance $\hat{s}_t$ for the current ongoing action at time t. If the regressed result $\hat{s}_{t-1}$ at the previous time step $(t-1)$ indicates that Layer #3 corresponds to the most *proper* window scale (i.e., $l_t^p = 3$), then our network will use Layers #1-3 for class prediction, while the activations from the layers above #3 are dropped (marked with *cross* in the figure). In this figure, we only show a subset of convolutional nodes of our SSNet, and other ones in the hierarchical structure (depicted as the solid lines in Figure 2) are omitted for clarity. The parameters of the convolutional layers and FC (fully connected) layers in our SSNet are trained jointly in an end-to-end fashion.

1, 2, 4, 8, ... for Layers #1, #2, #3, #4, ..., respectively. This design results in an exponential expansion of the perception scale across network layers. For example, the convolutional operation node $\mathbb{C}(t,3)$ in Figure 2 corresponds to a large scale of temporal window (8 frames: $[t-7,t]$), while the perception temporal window of $\mathbb{C}(t,2)$ is $[t-3,t]$ (4 frames). Here $\mathbb{C}(t,l)$ denotes the output activation of the dilated convolutional node in Layer #l ($l \in [1,\mathcal{L}]$) at time t, and $\mathcal{L}$ is the number of convolutional layers in our network. Note that any frame in the window $[t-7,t]$ can be perceived by the node $\mathbb{C}(t,3)$ with the hierarchical structure. This shows how the field of view expands over the layers in our network, while the resolution of input is not changed.

3.2. Scale Selection Scheme

In the streaming sequences, we can use the frames in a temporal window $[t-s,t]$ (with scale s) to perform action prediction for the time step t. However, finding a proper temporal scale s for different steps and inputs is not easy. At the early stages of an action, a relatively small scale is preferred, because larger windows can involve too many frames from the previous action, which interfere the recognition. On the contrary, if a large ratio of the action is observed (especially when the duration of this action is long), to obtain a reliable prediction, we need a larger s to cover more of its observed parts. This implies the importance of finding a proper scale value at each time step, rather than using a fixed scale at all steps.

In this section, we propose a scale selection scheme for online action prediction. The core idea is to regress a *proper* window scale at each time step, which will be used in its next step, in which the network can use this value to choose the *proper* layer for action prediction. At each step, as shown in Figure 2, the class label ($\hat{c}_t$) of the current action is predicted, and the temporal distance ($\hat{s}_t$) between the current action's start point and the current frame is also regressed. This distance indicates the performed part of the current action is assumed to be $[t-\hat{s}_t,t]$ at step t.

Assume we have obtained the regression result $\hat{s}_{t-1}$ at step $(t-1)$, thus at frame t, our network selects the time range $[(t-1)-\hat{s}_{t-1},t]$ for action prediction. Specifically, in our network design, the nodes in different layers correspond to different perception window scales, thus we can select the node from the *proper* layer to cover the performed part of the current action. For this *proper* layer l, we make sure its perception window's scale equals to (or slightly larger than) $\hat{s}_{t-1}+1$, while the perception window of its previous layer $(l-1)$ is smaller than $\hat{s}_{t-1}+1$. For example, Layer #2 in Figure 1 is the *proper* layer in this case.

We use l_t^p to denote the selected *proper* layer at step t. Then we aggregate the activations of the nodes $\mathbb{C}(t,l)$ ($l \in [1,l_t^p]$) in our network to generate a comprehensive representation for the selected time range as:

$$G_t^c = \frac{1}{l_t^p}\sum_{l=1}^{l_t^p}\mathbb{C}(t,l) \tag{1}$$

Note that we connect multiple layers ($[1,l_t^p]$) together to compute G_t^c, rather than using l_t^p only. This skip connection design can speed up convergence and enables the training

of much deeper models, as shown by [20, 21, 60]. Besides, it can also help to improve the representation capacity of our network, as the information from multiple layers corresponding to multiple scales is fused for current action. Finally, G_t^c is fed to the fully connected layers followed by a softmax classifier to predict the class label ($\hat{c}_t$) for the current time step. As shown in Figure 3, beside predicting the action class ($\hat{c}_t$), our network also generates a representation (G_t^s) to regress the start point's distance ($\hat{s}_t$):

$$G_t^s = \frac{1}{\mathcal{L}} \sum_{l=1}^{\mathcal{L}} \mathbb{C}(t, l) \tag{2}$$

For distance regression, we directly use the top convolutional layer $\mathcal{L}$ (*together with all the layers below it*), which has a large perception window (generally larger than the complete execution time of one action), rather than dynamically selecting a layer as in Eq (1). This is due to the essential difference between the regression task and the action label prediction task. Start point's distance regression can be regarded as regressing the position of the bonding [45] between the *current action* and its previous activities, thus involving information from the previous activity will not reduce (or even benefit) the regression performance for current action. Using Eq (2) also implies the distance regression is performed independently at each time step, and is not affected by the regression results of previous steps.

In object detection domain [39], such as Fast-RCNN [17], the bounding box of the current object was shown to be accurately regressed by a learning scheme. Similarly, the proposed network learns to regress the bounding (start point) of the current ongoing action reliably.

The regression result produced by the previous step ($t-1$) is used to guide the scale selection (with scale $\hat{s}_{t-1}+1$) for action prediction at the current step t. An alternative method can be: first regressing the scale $\hat{s}_t$ at step t, then using the scale $\hat{s}_t$ to directly perform action prediction for the same step t. We observe these two choices perform similarly in practice. This is intuitive as $\hat{s}_{t-1}+1$ is close to $\hat{s}_t$. The main difference of these two choices is the scale used at the beginning of a new action, because if we use the scale regressed by its previous step, the scale used at this step may be derived from the previous action, which is not proper. However, at the beginning frame of an action, too little information of the current action is observed, which makes prediction at this step very difficult even using the proper scale (only one frame), thus these two choices still perform similarly at this step. In the following frames, since more information is observed and proper scales can be used, both choices perform reliably. The framework will be less efficient if regressing for the same step, as the two tasks (regression and prediction) need to be conducted as two sequential stages at each time step (cannot be performed simultaneously).

3.3. Details of Network Structure

Our SSNet has 14 dilated convolutional layers. Specifically, we stack two similar sub-networks with dilation degrees (d) : $1, 2, 4, 8, ..., 64$ over the layers of each sub-network, i.e., the dilation degree (d) is reset to 1 at the beginning of each sub-network, as shown in Table 1. The motivation of this design is to achieve more variation for window scales (we obtain 14 different scales from 2 to 255 here). Besides, each sub-network can be intuitively regarded and implemented as a large convolutional module. Moreover, such a structure still guarantees each layer to perceive all the frames in its perception window without losing input resolutions. With such a design, the perception window scale of the top layer in our network is 255 frames, which covers more than 8-second sequence at the frame rate of common video cameras like Kinect. Generally, the duration of a full single action in most existing datasets is less than 8 seconds. Thus, the scale 255 is large enough for action analysis. Even if the whole duration time of an action is longer than 8 seconds, we believe the classification can be performed reliably when such a long segment (8 seconds) of the action has been perceived.

3.4. Activation Sharing Scheme

The proposed framework can be implemented in a very computation-efficient way. Though both action label prediction and distance regression are conducted on various window scales at each step, all of the computational steps are encapsulated in a single network with a hierarchical structure, i.e, we do not need separated networks or multiple scanning passes for action prediction at each step. In addition, although convolutional operations are performed over a sliding window at each step, the redundant computation of overlapping regions among different sliding positions are avoided. This is due to the design of causal convolution in our network, and many features (activations of convolution operations) computed in previous steps can be reused by the latter steps, which avoids redundant computation.

As shown in Eqs (1) and (2), at time step t, the prediction and regression are based on the nodes $\mathbb{C}(t, l)$, $l \in [1, l_t^p]$ or $l \in [1, \mathcal{L}]$. Each node $\mathbb{C}(t, l)$ is calculated based on only two input nodes, $\mathbb{C}(t-d_l, l-1)$ and $\mathbb{C}(t, l-1)$, as depicted in Figure 2. $\mathbb{C}(t-d_l, l-1)$ has already been computed at time step $t-d_l$. Therefore, to obtain $\mathbb{C}(t, l)$, we only need to calculate the activation of $\mathbb{C}(t, l-1)$. Similarly, $\mathbb{C}(t, l-1)$ can be computed after we get $\mathbb{C}(t, l-2)$.

Though we input a window of frames to SSNet at each step (t), we only need to calculate the activations of the nodes in column t of Figure 2, and all other convolution operations in the hierarchical structure can be copied from the previous time steps. This activation sharing makes our network efficient enough to be used in real-time applications.

Table 1. Details of the network structure.

Layer index	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Dilation degree (d)	1	2	4	8	16	32	64	1	2	4	8	16	32	64
Perception window scale	2	4	8	16	32	64	128	129	131	135	143	159	191	255
Output channels	50	50	50	50	50	50	50	50	50	50	50	50	50	50

3.5. Objective Function

The objective function of our network is formulated as:

$$\ell = \ell_c(\hat{c}_t, c_t) + \gamma \ell_s(\hat{s}_t, s_t) \quad (3)$$

where c_t is the ground truth class label, and s_t is the ground truth distance between the start point of the action and the current frame. γ is the weight for the regression task. ℓ_c is the negative log-likelihood loss measuring the difference between the true class label c_t and the predicted result $\hat{c}_t$ at time step t. ℓ_s is the regression loss defined as $\ell_s(\hat{s}_t, s_t) = (\hat{s}_t - s_t)^2$. Our objective function is minimized by stochastic gradient descent.

To train the network, we generate fixed-length clips from the annotated long sequences with sliding temporal windows. The length of each clip is equal to the perception scale of the top convolutional layer (255 frames). Each clip can then be fed to the SSNet. The class prediction is performed using the proper layer, which is chosen based on the ground truth distance of the start point, in the training phase.

During testing, action prediction is performed frame-by-frame through a sliding window, and the proper layer for prediction at each time step is determined by the output distance regression of its previous step. The ground truth information of the start point is not used in the testing phase.

4. Experiments

We evaluate our method on two challenging datasets: the OAD dataset [38] and the PKU-MMD dataset [40]. In both datasets, multiple action instances are contained in each long video. Beside the predefined action classes, these datasets also contain frames which belong to the background activity, thus we add a blank class to represent the frames in this situation. We conduct experiments with the following architectures:

(1) **SSNet.** This is our proposed model for 3D action prediction, which can select a proper layer to cover the performed part of the current ongoing action at each time step *by using the start point regression result*.

(2) **FSNet ($\mathcal{S}$).** Fixed Scale Network (FSNet) is similar to SSNet, but the action prediction is directly performed using the top layer. This indicates scale selection scheme is not used, and the prediction is based on a fixed window scale ($\mathcal{S}$) at all steps. We configure the structure to generate a set of FSNets, such that they have different perception window scales at the top layer. Concretely, five FSNets with different fixed scales ($\mathcal{S} = 15,\ 31,\ 63,\ 127,\ 255$) are evaluated. Note that to make a fair comparison, skip connections (see Eq (1)) are also used in each FSNet, i.e., all layers (corresponding to different scales) in a FSNet are connected as Eq (1) for action prediction at each step.

(3) **FSNet-MultiNet.** This baseline is a combination of multiple FSNets. A set of FSNets with different scales ($\mathcal{S} = 15,\ 31,\ 63,\ 127,\ 255$) are used for each time step. We then fuse the results of them, i.e., multi-scale multi-pass design is used to perform action prediction.

(4) Beside the aforementioned models, we also set an "ideal" baseline, **SSNet-GT**. Action prediction in SSNet-GT is also performed at the selected layer. However, we do not use the regression result to select the scale, instead, we directly use the *ground truth (GT)* distance of the start point to select the layer for action prediction at each step.

Our model is also compared to other state-of-the-art networks for 3D activity analysis:

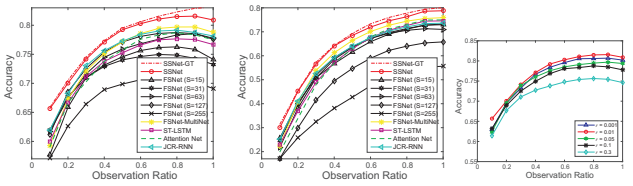
(1) **JCR-RNN** [38]. This network is a variant of LSTM, which models the context dependencies in temporal dimension of the untrimmed sequences. It obtains state-of-the-art performance of action detection in skeleton sequences on some benchmark datasets. A prediction of the current action class is provided at each frame of the streaming sequence.

(2) **ST-LSTM** [42]. This network achieves superior performance on 3D action recognition task. We adapt it to our online 3D action prediction task and generate a prediction of the action class at each frame of the streaming sequence.

(3) **Attention Net** [44]. This network adopts an attention mechanism to dynamically assign weights to different frames and different skeletal joints for 3D skeleton based action classification. A prediction of the action class is produced at each time step.

4.1. Implementation Details

Our experiments are conducted with the Torch7 toolbox [6]. We train the network from scratch, i.e., network parameters are initialized with small random numbers (uniform distribution in [-0.08, 0.08]). The learning rate, momentum, and decay rate are set to 10^{-3}, 0.9, and 0.95, respectively. Residual connections [20] are used over different convolutional layers. GLU [8] is the activation function used for the convolution operations in our network. The output dimensions of FC1, FC3, FC4, and FC5 in Figure 3 are 50, 50, 50, and 1, respectively. FC2's output dimension is determined by the class number of each specific dataset. In our experiment, γ in Eq (3) is set to 0.01. The above-mentioned parameters are obtained by using cross validation protocol on the training sets, and the parameter set giving the optimum performance is adopted.



(a) (b) (c)

Figure 4. Prediction results on (a) OAD and (b) PKU-MMD datasets. Fig (c) shows comparison of different γ values on OAD.

Due to the dilated convolutional design, the number of parameters for SSNet is relatively small ($\sim$270K). We train our network on a single NVIDIA GTX-1080 GPU. We evaluate the efficiency of our method for online prediction in continuous sequence, and the speed reaches 70 *fps*, which indicates that our framework responds extremely fast.

4.2. Experiments on the OAD Dataset

The OAD dataset [38] was captured with Kinect v2 in daily-life indoor environments. It includes 10 actions. The long video sequences in this dataset correspond to about 700 action instances. The starting and ending frames of each action are annotated in this dataset. 30 long sequences are used for training, and 20 long sequences are for testing.

We report the prediction results in Figure 4(a) and Table 2(a). In the figures and tables, the accuracy of an observation ratio $p\%$ denotes the average accuracy of the predictions in the observed segment ($p\%$) of the action instance.

Note that the special baseline SSNet-GT performs action prediction with the *Ground Truth* scale at each step, thus it provides the best results. Our SSNet with regressed scale even achieves comparable results to this "ideal" baseline (SSNet-GT), which indicates the effectiveness of our scale selection scheme for online action prediction at each progress level. Apart from the "ideal" SSNet-GT model, our proposed SSNet yields the best prediction results among all methods at all observation ratios. Specifically, our SSNet can even produce a quite reliable prediction (about 66% accuracy) at the early stage when only a small ratio (10%) of the action instance is observed.

The performance of our SSNet is much better than FSNets which perform prediction with fixed-scale windows at each time step. Even fusing a set of FSNets with different scales, FSNet-MultiNet is still weaker than our single SSNet at all progress levels. This demonstrates that our proposed scale selection scheme, which guides the SSNet to dynamically cover the performed part of current action at each step, is very effective for online action prediction.

We also find our SSNet significantly outperforms the state-of-the-art RNN/LSTM based methods, JCR-RNN [38] and ST-LSTM [42] which can handle continuous skeleton sequences. The performance disparity can be explained as: (1) At the early stages (eg. 10%), our SSNet can focus on the performed part of current action by using the selected scale, while RNN models [38, 42] may bring information from the previous actions which can interfere the prediction for current action. (2) At the latter stages (eg. 90%), the context information from the early part of current action may vanish in RNN model with its hidden state evolving frame by frame, while our SSNet, which uses convolutional layers to model the temporal dependencies over the frames, can still handle the long-term context dependency information in the temporal window. (3) There are hierarchical structures of the motion patterns in temporal dimension [58]. However, RNN/LSTM models do not have strong ability in modeling this structure. Our SSNet also outperforms the Attention Net [44] which assigns weights to differen frames and joints. This indicates the superiority of our SSNet with explicit scale selection.

4.3. Experiments on the PKU-MMD Dataset

PKU-MMD [40] is a large dataset for 3D activity analysis in continuous sequences. Cross-subject evaluation protocol is used, in which 57 subjects are used for training, and the remaining 9 subjects are for testing. Considering the large amount of data, we use the videos which contain the challenging interaction actions for our experiment, and sample 1 frame from every 4 frames for these videos. Our method achieves the best results at all progress levels, as shown in Figure 4(b) and Table 2(b). Our SSNet outperforms other methods significantly, even when only a very small ratio (10%) of the action is observed. This indicates our method can produce a more reliable prediction at the early stage by focusing on the current action, compared to other methods which do not explicitly consider scale selection. We also find FSNet with fixed scale at each step is sensitive to the scale used, as different scales provide very different results. This further demonstrates that our SSNet, which dynamically chooses the proper scale at each step to perform prediction, is effective for online action prediction.

4.4. More Experiments

Evaluation of distance regression. We adopt the metric $SL\text{-}Score$ proposed in [38] to evaluate the performance of distance regression of our network, which is calculated as $e^{-|\hat{s}-s|/d}$, where s and $\hat{s}$ are respectively the ground truth distance and regressed distance to the action's start point, and d is the length of an action. For false classification samples, the score is set to 0. Note that in our SSNet, start point regression is performed based on the top convolutional layer (see Eq. (2)). We also test an alternative model (we call it "SSNet*" here), in which the distance regression is performed at the selected layer, which is determined by the regression result of the previous time step (similar to Eq. (1)). We report the regression performance of SSNet and SSNet* in Table 3. As the action detection method, JCR-RNN [38],

Table 2. Performance comparison of prediction accuracies. The last column SSNet-GT is an "ideal" baseline, in which *Ground Truth* scales are used for action prediction. We observe our SSNet, which performs prediction with regressed scales, is even comparable to SSNet-GT.

Observation Ratio	ST-LSTM	Attention Net	JCR-RNN	FSNet (15)	FSNet (31)	FSNet (63)	FSNet (127)	FSNet (255)	FSNet-MultiNet	SSNet	SSNet-GT
(a) Prediction accuracies on **OAD** dataset. *Refer to Figure 4(a) for more results.*											
10%	60.0%	59.0%	62.0%	57.7%	62.0%	61.7%	61.2%	57.1%	59.3%	**65.6%**	65.8%
50%	75.3%	75.8%	77.3%	74.6%	74.0%	75.9%	77.1%	69.9%	77.2%	**79.2%**	79.5%
90%	77.5%	78.3%	78.8%	75.9%	74.3%	78.6%	78.5%	70.2%	79.7%	**81.6%**	82.9%
(b) Prediction accuracies on **PKU-MMD** dataset. *Refer to Figure 4(b) for more results.*											
10%	22.9%	19.8%	25.3%	25.6%	24.6%	21.8%	17.1%	17.1%	22.4%	**30.0%**	31.4%
50%	63.0%	62.9%	64.0%	61.6%	63.8%	63.6%	54.7%	42.7%	66.4%	**68.5%**	69.3%
90%	74.5%	74.9%	73.4%	72.8%	72.8%	71.3%	65.4%	55.2%	75.5%	**78.6%**	79.4%

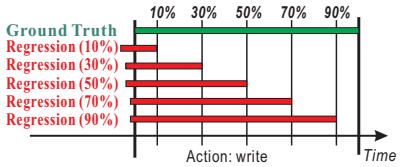


Figure 5. An example of the regression results.

also estimates the start point, we also compare our method with it. The results show that SSNet provides the best regression performance. We also observe the regression result of SSNet is better than SSNet*. A possible explantation is: the start point can be seen as the bounding between the current action and the previous activities, thus we do not need to select a layer to only focus on the observed part of current action to perform regression, and the information from the previous action can even help in start point regression. Thus we perform regression by directly using the top layer.

We also evaluate the average regression errors in the observed segment ($p\%$) in Table 4 on PKU-MMD dataset. The regression error is calculated as $|\hat{s} - s|$. We find our method regresses the distance reliably. When only a small ratio (3%) of the action instance has been observed, the average regression error is 7 frames. The regression becomes more reliable if more frames (eg, 10%) are observed.

We also show an example of results in Figure 5. It shows that our SSNet achieves promising regression performance.

Evaluation of network configurations. We configure the maximum dilation degree and the layer number to generate a set of SSNets, which have different maximum perception window scales at the top layers. The results in Table 5 show that using more layers are beneficial for performance as the perception window scale of top layer increases, but the performance of 16 layers is almost the same as 14 layers. A possible explanation is that the duration time of most actions is less than 255 frames. Besides, 255 frames are long enough for action analysis. Thus using the SSNet with 14 layers (maximum window scale 255) is suitable.

We also evaluate the performance of our SSNet with different γ values (see Eq (3)) in Figure 4(c). We observe our SSNet yields the best performance when γ is set to 0.01.

Frame-Level Classification. As action classification is performed at each frame, the average classification accuracies over all frames are also evaluated, as shown in Table 6.

Table 3. Start point regression performance ($SL\text{-}Score$).

Dataset	JCR-RNN [38]	SSNet*	SSNet
OAD	0.42	0.61	**0.69**
PKU-MMD	0.61	0.67	**0.72**

Table 4. Start point regression errors.

Observed Segment	3%	6%	10%	40%	70%	100%
Error (frames)	7	4	3	3	3	3

5. Conclusion

We have proposed the SSNet for online 3D action prediction. A stack of convolutional layers are introduced to model the dependencies in temporal dimension. A scale selection scheme is also proposed for SSNet, with which the network can choose a proper layer corresponding to the most proper scale for action prediction at each step. SSNet shows superior performance on the evaluated benchmark datasets.

Acknowledgement

This research was carried out at Rapid-Rich Object Search (ROSE) Lab at Nanyang Technological University. ROSE is supported by National Research Foundation, Singapore, under IDM Strategic Research Programme. This work is in part supported by National Natural Science Foundation of China (61661146005, U1611461, 61390515). We thank NVIDIA AI Technology Centre for GPU donation.

Table 5. Evaluation of different configurations of SSNet on OAD.

Number of conv. layers	8	10	12	14	16
Max. dilation degree	8	16	32	64	128
Max. perception wind. scale	*31*	*63*	*127*	*255*	*511*
Regression ($SL\text{-}Score$)	0.62	0.66	0.68	0.69	0.69
Prediction accuracy (%)	73.4	76.9	78.0	78.7	78.7

Table 6. Frame-level classification accuracies. FSNet (best) denotes the FSNet which gives the best results among all FSNets.

Dataset	[42]	[44]	[38]	FSNet (best)	SSNet
OAD	0.77	0.75	0.79	0.79	**0.81**
PKU-MMD	0.78	0.80	0.79	0.80	**0.82**

References

[1] J. K. Aggarwal and L. Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 2014.

[2] S. Baek, K. I. Kim, and T.-K. Kim. Real-time online action detection forests using spatio-temporal contexts. In *WACV*, 2017.

[3] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *CVPRW*, 2012.

[4] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, and S. Wang. Recognize human activities from partially observed videos. In *CVPR*, 2013.

[5] G. S. Chambers, S. Venkatesh, G. A. West, and H. H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *ICPR*, 2002.

[6] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *NIPSW*, 2011.

[7] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *ICCV*, 2017.

[8] Y. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017.

[9] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018.

[10] Y. Du, Y. Fu, and L. Wang. Skeleton based action recognition with convolutional neural network. In *ACPR*, 2015.

[11] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[12] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *T-PAMI*, 2009.

[13] G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads: Human action recognition using joint quadruples. In *ICPR*, 2014.

[14] J. Gao et al. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017.

[15] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[16] J. Gao, Z. Yang, and R. Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017.

[17] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[18] J. Gu, G. Wang, J. Cai, and T. Chen. An empirical study of language cnn for image captioning. In *ICCV*, 2017.

[19] F. Han, B. Reily, W. Hoff, and H. Zhang. Space-time representation of people based on 3d skeletal data: a review. *CVIU*, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[22] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 2014.

[23] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai. Real-time rgb-d activity prediction by soft regression. In *ECCV*, 2016.

[24] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *CVPR*, 2017.

[25] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 1973.

[26] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid. Skeletonnet: Mining deep part features for 3-d action recognition. *SPL*, 2017.

[27] Q. Ke, M. Bennamoun, S. An, F. Boussaid, and F. Sohel. Human interaction prediction using deep temporal features. In *ECCV*, 2016.

[28] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. Leveraging structural context models and ranking score fusion for human interaction prediction. *TMM*, 2017.

[29] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, 2017.

[30] T. S. Kim and A. Reiter. Interpretable 3d human action analysis with temporal convolutional networks. *arXiv*, 2017.

[31] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *ECCV*, 2014.

[32] Y. Kong, Z. Tao, and Y. Fu. Deep sequential context networks for action prediction. In *CVPR*, 2017.

[33] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. *arXiv*, 2016.

[34] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 1995.

[35] B. Li, H. Chen, Y. Chen, Y. Dai, and M. He. Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network. *arXiv*, 2017.

[36] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *T-PAMI*, 2014.

[37] K. Li, J. Hu, and Y. Fu. Modeling complex temporal composition of actionlets for activity prediction. In *ECCV*, 2012.

[38] 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 *ECCV*, 2016.

[39] T.-Y. Lin et al. Feature pyramid networks for object detection. *CVPR*, 2017.

[40] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv*, 2017.

[41] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang. Skeleton-based action recognition using spatio-temporal l-stm network with trust gates. *TPAMI*, 2017.

[42] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016.

[43] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *TIP*, 2018.

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

[45] S. Liu, L. Feng, Y. Liu, H. Qiao, J. Wu, and W. Wang. Manifold warp segmentation of human action. *TNNLS*, 2017.

[46] Q. Ma, L. Shen, E. Chen, S. Tian, J. Wang, and G. W. Cottrell. Walking walking walking: Action recognition from action echoes. In *IJCAI*, 2017.

[47] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006.

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

[49] L. L. Presti and M. La Cascia. 3d skeleton-based human action classification: a survey. *Pattern Recognition*, 2016.

[50] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

[51] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016.

[52] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang. Multimodal multipart learning for action recognition in depth videos. *T-PAMI*, 2016.

[53] A. Sharaf, M. Torki, M. E. Hussein, and M. El-Saban. Real-time multi-scale action detection from 3d skeleton data. In *WACV*, 2015.

[54] Z. Shou et al. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.

[55] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[56] P. Siva and T. Xiang. Weakly supervised action detection. In *BMVC*, 2011.

[57] 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 *AAAI*, 2017.

[58] Y. Song, L.-P. Morency, and R. Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, 2013.

[59] L. Tian, M. Li, Y. Hao, J. Liu, G. Zhang, and Y. Q. Chen. Robust 3-d human detection in complex environments with a depth camera. *TMM*, 2018.

[60] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, 2016.

[61] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *TPAMI*, 2017.

[62] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *ICCV*, 2015.

[63] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014.

[64] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3d human action recognition. *T-PAMI*, 2014.

[65] L. Wang et al. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.

[66] P. Wang, Z. Li, Y. Hou, and W. Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *ACM MM*, 2016.

[67] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *ECCV*, 2016.

[68] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu. Concurrent action detection with structural prediction. In *ICCV*, 2013.

[69] J. Weng, C. Weng, and J. Yuan. Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In *CVPR*, 2017.

[70] L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, 2012.

[71] Z. Xu, L. Qing, and J. Miao. Activity auto-completion: Predicting human activities from partial videos. In *ICCV*, 2015.

[72] X. Yang and Y. Tian. Effective 3d action recognition using eigenjoints. *JVCIR*, 2014.

[73] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[74] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[75] G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *ACCV*, 2014.

[76] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[77] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013.

[78] G. Zhang, J. Liu, H. Li, Y. Q. Chen, and L. S. Davis. Joint human detection and head pose estimation via multistream networks for rgb-d videos. *SPL*, 2017.

[79] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang. Rgb-d-based action recognition datasets: A survey. *PR*, 2016.

[80] P. Zhang et al. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. *arXiv*, 2017.

[81] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *WACV*, 2017.

[82] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, D. Lin, and X. Tang. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[83] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016.

[84] Y. Zhu and S. Newsam. Efficient action detection in untrimmed videos via multi-task learning. In *WACV*, 2017.