Skeleton-Based Online Action Prediction Using Scale Selection Network

Jun Liu[1] jliu029@ntu.edu.sg
Amir Shahroudy[2] amirsh@chalmers.se
Gang Wang[3] gangwang6@gmail.com
Ling-Yu Duan[4] lingyu@pku.edu.cn
Alex C. Kot[1] eackot@ntu.edu.sg

[1] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2] Chalmers University of Technology, Sweden
[3] Alibaba Group, China
[4] Peking University, China

Abstract

Action prediction is to recognize the class label of an ongoing activity when only a part of it is observed. 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 temporal axis. Since there are significant temporal scale variations in the observed part of the ongoing action at different time steps, a novel window scale selection method is proposed to make our network focus on the performed part of the ongoing action and try to suppress the possible incoming interference from the previous actions at each step. An activation sharing scheme is also proposed to handle the overlapping computations among the adjacent time steps, which enables our framework to run more efficiently. Moreover, to enhance the performance of our framework for action prediction with the skeletal input data, a hierarchy of dilated tree convolutions are also designed to learn the multi-level structured semantic representations over the skeleton joints at each frame. Our proposed approach is evaluated on four challenging datasets. The extensive experiments demonstrate the effectiveness of our method for skeleton-based online action prediction.

1. Introduction

In action prediction (early action recognition), the goal is to predict the class label of an ongoing action from an observed part of it over temporal axis so far. 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 research attention due to its wide range of applications in security surveillance, human-machine interaction, patient monitoring, etc [1, 2]. Most of the existing works [1, 3, 4] focus on action prediction in well-segmented videos, for which each video contains exactly one action instance. However, in more practical scenarios, such as online human-machine interaction systems, plenty of unsegmented action instances are contained in a streaming sequence. In this paper, we address this challenging task: "online action prediction in untrimmed video", *i.e.*, we aim to recognize the current ongoing action from the observed part of it at each time step of the data stream, which can include multiple actions, as illustrated in Figure 1(a).

The biological studies [5] demonstrate that skeleton data is informative enough for representing human behavior, even without appearance information [6]. Human activities are naturally performed in 3D space, thus 3D skeleton data is suitable for representing human actions [7]. The 3D skeleton information can be easily and effectively acquired in real-time with the low-cost depth sensors [8], such as Microsoft Kinect and Asus Xtion. As a result, activity analysis with 3D skeleton data becomes a popular domain of research [9–17] thanks to its succinctness, high level representation, and robustness against variations in viewpoints, illumination, clothing textures, and background clutter [1, 18, 19].

We investigate real-time action prediction with the continuous 3D skeleton data in this paper. To predict the class label of the current ongoing action at each time step, we adopt a sliding window over the temporal axis of the input streams of skeleton sequences, and the frames under the window are used as input to perform action prediction.

The sliding window design has been widely employed for a series of vision related tasks, such as object recognition [20], pedestrian detection [21], activity detection [22–25],

etc. Most of these works utilize 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. This makes it difficult to determine the scale of the sliding window.

The untrimmed streaming sequence may contain multiple action instances, as shown in Figure 1(a). The order of the actions can be arbitrary, and the duration of different instances is 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 instance, at an early temporal stage, it is beneficial to use a relatively smaller temporal window, because the larger window sizes may include frames from the previous action instances which can mislead the recognition of the current instance. Conversely, if a large part of the current action has already been 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 tackle the aforementioned challenges, in this paper, a novel Scale Selection Network (SSNet) is proposed for online action prediction. Instead of using a fixed scale or multi-scale multi-pass scans at each time step, we supervise our network to dynamically learn the proper temporal window scale at each step to cover the performed part of the current action instance. In our approach, 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 temporal step (next frame), we can utilize this value as the temporal window scale for action class prediction.

In our network, we apply convolutional analysis in temporal dimension to model the motion dynamics over the frames for skeleton-based action prediction. A hierarchical architecture with dilated convolution filters 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 the current action and try to suppress the unrelated data from the previous ones. Hence it produces 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 approaches that utilize sliding window designs, the computational efficiency is often relatively low due to the overlapping design and exhaustive multi-scale multi-round scans. In our method, the action prediction is performed with a regressed scale at each step, which avoids multi-pass scans. 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.

In addition, to improve the performance of our network in handling the 3D skeleton data as input, we also propose a hierarchy of dilated tree convolutions to learn the multi-level structured semantic representations over the skeleton joints at each frame for our action prediction network.

The main contributions of this paper are summarized as follows:

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. Our proposed SSNet 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 learn the proper temporal scale at each step, such that the network can mainly focus on the performed part of the current action, and try to avoid the clutter from the previous actions data in the input online stream.
3. A hierarchy of dilated tree convolutions are also proposed to learn multi-level structured representations for the input skeleton data and improve the performance of our SSNet for skeleton-based action prediction.
4. The proposed framework is very efficient for online action analysis thanks to the computation sharing over different time steps.
5. We perform action prediction with our SSNet which is end-to-end trainable, rather than using expensive multi-stage multi-network design at each step.
6. The proposed method achieves superior performance on four challenging datasets for 3D skeleton-based activity analysis.

The remainder of this paper is organized as follows. We review the related works in section 2. In section 3, we introduce our proposed SSNet for skeleton-based online action prediction in detail. We present the experimental results and comparisons in section 4. Finally, we conclude the paper in section 5.

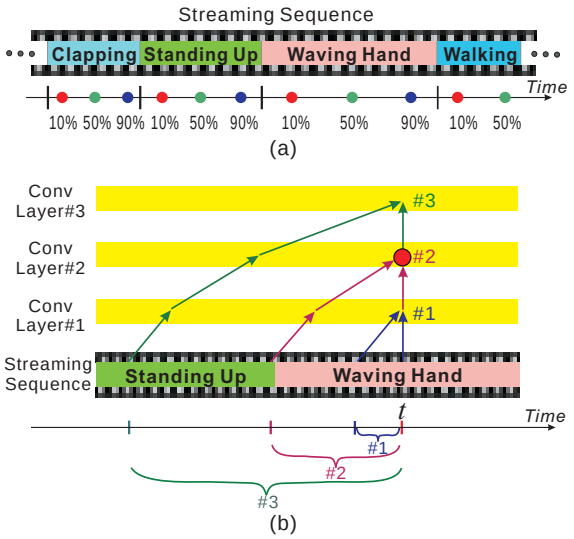


Figure 1: Figure (a) illustrates an untrimmed streaming sequence that contains multiple action instances. We need to recognize the current ongoing action at each time step when only a part (*e.g.*, 10%) of it is performed. Figure (b) depicts our SSNet approach for online action prediction. At time t, 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 time step t.

2. Related Work

Skeleton-Based Action Recognition. With the advent of cheap and easy-to-use depth sensors, such as Kinect [8], 3D skeleton-based human action recognition became a popular research domain [26, 27], and a series of hand-crafted features [28–37] and deep learning-based approaches [18, 19, 38–44] have been proposed.

Most of the existing skeleton-based action recognition methods [30, 40, 45–48] receive the fully observed segmented videos as input (each sample contains one full action instance), and derive a class label. The proposed skeleton-based online 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 only partly observed. There are a limited number of skeleton-based action recognition methods [49] for untrimmed online sequences. Different from these works, the proposed SSNet framework predicts the class label of the current ongoing action by utilizing its predicted observation ratio.

Action Prediction. Predicting (recognizing) an action before it gets fully performed has attracted a lot of research attention recently [2, 3, 50–54].

Cao *et al*. [2] formulated the prediction task as a posterior-maximization problem, and applied sparse coding for action prediction. Ryoo *et al*. [51] represented each action as an integral histogram of spatio-temporal features. They also developed a recognition methodology called dynamic bag-of-words (DBoW) for activity prediction. Li *et al*. [55] designed a predictive accumulative function. In their method, the human activities are represented as a temporal composition of constituent actionlets. Kong *et al*. [50] proposed a discriminative multi-scale model for early action recognition. Ke *et al*. [3] extracted deep features in optical flow images for activity prediction.

Hu *et al*. [1] explored to incorporate 3D skeleton information for real-time action prediction in the *well-segmented* sequences, *i.e*., each sequence includes only one action. They introduced a soft regression strategy for action prediction. An accumulative frame feature was also designed to make their method work efficiently. However, their framework is not suitable for online action prediction in the *untrimmed* continuous skeleton sequence that contains multiple action instances.

Action Analysis with Untrimmed Sequences. Beside the online action prediction task, the problem of temporal action detection [22, 56, 56–66] also copes with untrimmed videos. Several methods attempted online detection [67], while most of the action detection approaches are developed for handling offline mode that conducts detection after observing the whole long sequence [22, 23, 68].

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 from its observed part so far, even when only a small ratio of it is performed.

Sliding window-based design [24, 25, 61, 69] and action proposals [60] have been adopted for action detection. Zanfir *et al*. [24] used a sliding window with one fixed scale (obtained by cross validation) for action detection. Shou *et al*. [70] adopted multi-scale windows for action detection via multi-stage networks.

Differently, in our online action prediction task, determining the scale of the temporal window is challenging due to the scale variations of the observed part of the ongoing action. Also, rather than using one fixed scale [24] or multi-scale multi-round scans [70, 71], we propose a novel SSNet for online prediction, which is supervised to choose the proper window for prediction at each time step. Moreover, the redundant computations are efficiently shared over different steps in our approach.

This manuscript is the extension of our recent conference paper [72]. The contributions of this work over [72] are as follows. In [72], the coordinates of the skeleton joints at each frame were simply concatenated to form a vector rep-

resenting the current frame's pose. Such a representation ignores the underlying semantics of spatial pose structures. In this paper, we propose a hierarchy of dilated tree convolutions to process the input data and learn more powerful multi-level structured semantic representations at each frame of the streaming skeleton sequence. The newly proposed multi-level structured representation enhances the capability of our framework for action prediction in 3D skeleton streams. In addition, we provide a more in-depth description of the proposed method and its implementation details. Furthermore, we extensively evaluate the proposed action prediction framework on two more datasets, including the large-scale ChaLearn Gesture dataset for body language understanding [73] and the G3D dataset for gaming action analysis [49]. More extensive empirical analysis of the proposed approach is also provided in this paper.

3. The Proposed Method

We introduce the proposed network architecture, Scale Selection Network (SSNet), for skeleton-based online action prediction in this section. The overall schema of this method is illustrated in Figure 2. In the proposed network, the one dimensional (1-D) convolutions are performed in temporal domain to model the motion dynamics over the frames. The inputs of SSNet are the frames within a temporal window at each time step. In order to tackle the scale variations 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. To better deal with the input data modality, a hierarchy of dilated tree convolutions are also introduced to process the input skeleton data for our network.

3.1. Temporal Modeling with Convolutional Layers

Convolutional networks [74] have proven their superior strength in modeling the time series data [75–77]. For example, van den Oord *et al*. [75] proposed a convolutional model, called WaveNet, for audio signal generation, and Dauphin *et al*. [76] 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 skeleton-based action prediction task. Specifically, a hierarchical architecture with dilated convolutional layers is leveraged in our model to learn a comprehensive representation over the video frames within a temporal window.

Dilated convolution. The main building blocks of our network model are dilated causal convolutions. Causal design [75] enforces the prediction task at time t to be based on the available information before t (including t) without using the future information. Dilated convolution [78] applies the convolutional filter over a larger field than the filter's length, and some input values inside the field are skipped by a certain step size.

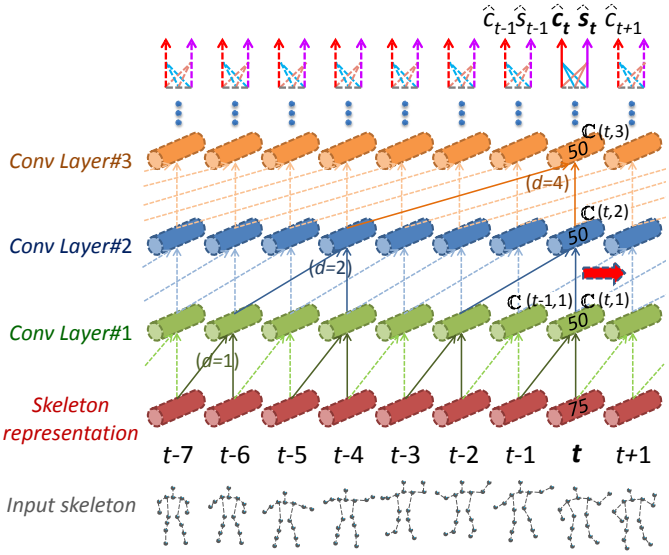


Figure 2: Illustration of the proposed SSNet for action prediction over the temporal axis. The solid lines denote the SSNet links activated at current step t, and the dashed lines indicate the links activated at other time steps. Our SSNet has 14 1-D convolutional layers. Here we only show 3 layers for clarity. At each time step, SSNet predicts the class ($\hat{c}_t$) of the ongoing action, and also estimates the temporal 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. (Best viewed in color)

Concretely, dilated convolution (also known as "convolution with holes") can be formulated as presented in [78]:

$$(X *_d w)(\mathbf{p}) = \sum_{\mathbf{t}+d\mathbf{s}=\mathbf{p}} X(\mathbf{t})\, w(\mathbf{s}) \qquad (1)$$

where $*_d$ indicates the dilated convolutional operation, X is the input, w is the filter, and d denotes the dilation rate of the convolution ($d = 1$ represents the standard convolution).

In order to show how the dilated convolution is used in our model, we illustrate the mechanism of a dilated convolutional layer in Figure 4.

As shown in Figure 4, at each position (*e.g.*, the position t), the dilated convolutional filter (with dilation rate d) works over two input time steps (t and $t - d$), and the other time steps between these two steps are not considered for the convolutional operation at this position. Let $\mathbb{C}(t, l)$ denote the activation of the convolutional node at the position

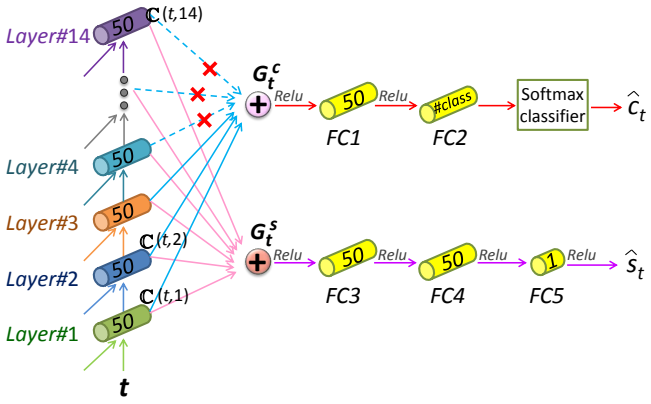


Figure 3: Details of our SSNet that jointly predicts the class label $\hat{c}_t$ and regresses the start point's distance $\hat{s}_t$ for the current ongoing action **at time** t. If the regressed result $\hat{s}_{t-1}$ at the previous time step $(t-1)$ indicates that layer #3 corresponds to the most *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.

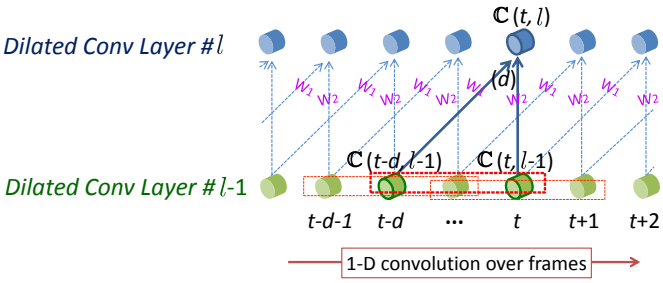


Figure 4: Illustration of the dilated convolution layer used in our network. At each position, the 1-D convolutional filter covers a time range (labeled as a red box), and only the two boundary nodes (corresponding to two time steps) in the covered range are used, while the other nodes between these two nodes are not used by the dilated convolutional operation for this position.

t in the dilated convolutional layer #l ($l \in [1, \mathcal{L}]$, and $\mathcal{L}$ denotes the number of 1-D convolutional layers in our network). Then $\mathbb{C}(t,l)$ can be calculated as:

$$\mathbb{C}(t,l) = f\Big(W_1\, \mathbb{C}(t-d, l-1) + W_2\, \mathbb{C}(t, l-1) + b\Big) \quad (2)$$

where $f(\cdot)$ is a non-linear activation function. W_1 and W_2 (together with the bias b) are the parameters of the dilated convolutional filter, which are shared at the same layer, as illustrated in Figure 4.

It is intuitive to use the aforementioned dilated convolution for human action analysis, because 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. In contrast, by configuring the dilation rate (d), dilated convolution can support expansion of the receptive field very efficiently, without bringing more parameters. In addition, it does not need any extra pooling operations, thus it can well maintain the ordering information of the inputs [78].

Multiple dilated convolutional layers. In our method, we stack multiple dilated convolutional layers, as illustrated in Figure 2. The dilation rate increases exponentially over the layers in our network, *i.e.*, we set d to $1,\ 2,\ 4,\ 8,\ \ldots$ for layers #1, #2, #3, #4, ..., respectively.

This design results in an **exponential** expansion of the perception scale across the network layers. For example, the perception temporal window of the convolutional operation node $\mathbb{C}(t,2)$ in layer #2 (see Figure 2) is $[t-3, t]$ (4 frames), while the node $\mathbb{C}(t,3)$ in layer #3 corresponds to a larger scale of temporal window (8 frames: $[t-7, t]$).

It is worth mentioning that all the video frames 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 coverage of the input is kept.

3.2. Scale Selection

For the streaming sequences, we can utilize the frames in a temporal window $[t-s, t]$ (with scale s) to perform action prediction at 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 may influence 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.

We propose a scale selection scheme for online action prediction in this section. The core idea is to regress a *proper* window scale at each time step, and then at the next time step, the network can use this scale value to choose the *proper* layers for action prediction.

At each step, as shown in Figure 2, the class label ($\hat{c}_t$)

Table 1: Details of the main structure of SSNet. Refer to Figure 13 for the detailed architecture configurations of SSNet.

1-D convolutional layer index	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Dilation rate (d)	1	2	4	8	16	32	64	1	2	4	8	16	32	64
Perception temporal window scale (frames)	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

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 that the performed part of the current action is assumed to be $[t - \hat{s}_t, t]$ at step t.

Assuming that 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 temporal 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.

Let l_t^p 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:

$$G_t^c = \frac{1}{l_t^p} \sum_{l=1}^{l_t^p} \mathbb{C}(t, l) \tag{3}$$

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 [79, 80]. Besides, it can also help to improve the representation capability 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 ($\hat{s}_t$):

$$G_t^s = \frac{1}{\mathcal{L}} \sum_{l=1}^{\mathcal{L}} \mathbb{C}(t, l) \tag{4}$$

For the distance regression, we directly adopt 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 (3). 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 [81] 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 (4) also implies the distance regression is performed independently at each time step, and is not affected by the regression results of the previous steps.

In the domain of object detection [82], such as Fast-RCNN [83], the bounding box of the current object was shown to be accurately regressed by a learning scheme. Similarly, our 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 the Main Structure

The proposed SSNet has 14 dilated convolutional layers for temporal modeling. Specifically, we stack two similar sub-networks with dilation rates $(d) : 1, 2, 4, 8, ..., 64$ over the layers of each sub-network, *i.e.*, the dilation rate (d) is reset to 1 at the beginning of each sub-network, as shown in

Table 1 and Figure 13. The motivation of this design is to achieve more variation for the temporal 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 design still guarantees the node at each layer to perceive all the video frames in its perception window (*i.e.*, without losing input coverage), due to the hierarchial structure of SSNet.

With such a design, the perception temporal window scale of the top layer in our network is 255 frames, which covers more than 8-second sequence at the recording 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 temporal 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.

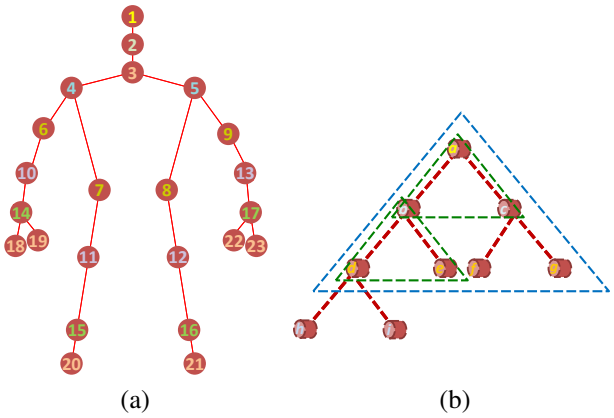
3.4. Activation Sharing Scheme

Our framework can be implemented in a very computation-efficient way. Although 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 (see Figure 2), *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 computations of the overlapping regions among different sliding positions are avoided. With the causal convolution design, many features (activations of convolutional operations) computed in previous steps can be reused by the latter steps, which avoids redundant computation.

As depicted in Eqs (3) and (4), 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 shown 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 $\mathbb{C}(t, l-2)$.

As a result, although we feed a window of frames to SSNet at each time step (t), we only need to calculate the activations of the nodes in column t of Figure 2, and all other convolutional 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.



(a) (b)

Figure 5: (a) The skeleton joints of the human body form a tree structure. We set the head joint (joint 1) as the root node, and the height of the tree in this figure is 8. (b) Illustration of the convolution with triangular filters sliding over the tree structure. The green and the blue triangles indicate the convolutions with two different filter sizes.

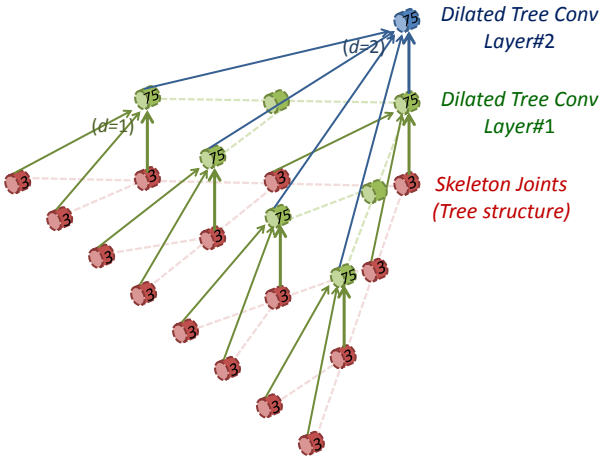


Figure 6: Illustration of the hierarchy of dilated tree convolutions that learns the multi-level structured representations over the input skeleton joints (labeled in red) at each frame. The solid arrows denote the dilated tree convolutions with triangular filters. In our method, 3 dilated tree convolutional layers are used to cover the input skeleton tree with height 8, while in this figure, we only show 2 layers that cover the tree with height 4 for clarity. Note that the bottom of this figure shows a full binary tree, while the human skeleton only has a subset of the nodes of a full binary tree. Therefore, in implementation, the convolutional operations only need to be performed on a subset of the nodes. The channel number of the input skeleton is 3, namely, the 3D coordinates (x, y, z) of each joint. (Best viewed in color)

3.5. Multi-level Structured Skeleton Representations

As mentioned above, in our framework, the streaming 3D skeleton data is fed to the SSNet. A naive way to perform action prediction with such an input data structure is to concatenate the 3D coordinates of all joints at each frame to form a vector (that we call it as coordinate concatenation representation). We can then feed this coordinate concatenation representation of each frame to the SSNet as input (see Figure 2). However, the semantic structure amongst the skeleton joints in a frame is ignored in this representation. As illustrated in Figure 5(a), the skeleton joints in every human body configuration are physically connected in a semantical tree structure in the spatial domain, and utilizing such structure information has shown to be quite helpful for human activity analysis [18, 19, 31].

Instead of directly using the method of coordinate concatenation, in this paper, we model the spatial tree structure of the skeleton joints, in order to capture the posture information of the human body more effectively at each frame and thus strengthen the capability of our framework in skeleton-based action prediction.

Specifically, we propose a hierarchy of dilated tree convolutions in spatial domain to learn the multi-level (local, mid-level, and holistic) structured representations for the tree structure of the skeleton in each frame. The proposed hierarchical dilated tree convolution for spatial domain modeling is essentially an extension of the multi-layer 1-D dilated convolution that is introduced in section 3.1 for temporal modeling. Below we introduce this design in detail.

Convolution over tree structure. Convolutional networks are powerful tools in modeling the spatial visual structures [74]. Here to model the discussed semantic structure of the human skeleton, we propose to apply convolutions by using triangular filters sliding over the nodes of the tree, as shown in Figure 5(b). At each step of the convolution, the triangular filter covers a sub-tree region, and the nodes in this region are used to produce an activation as a semantic representation of this position. This process is similar to the common convolutional operations that slide over the pixels of an input image or previous layer's feature maps. Different sizes of the triangular filters can also be used for this process, as shown in Figure 5(b).

In our method, zero padding is adopted for the convolution over the skeleton tree, *i.e*., if a certain node (*e.g*., joint 2 in Figure 5(a)) has only one child (joint 3), to perform convolution at this node position, we set this child (joint 3) as the left node, and its right node is filled with zero. Similarly, for the leaf nodes (*e.g*., joint 20), both of the child nodes are filled with zero.

A hierarchy of dilated tree convolutions. In order to learn representations that are effective and discriminative for representing the skeletal data in a frame, we stack multiple convolutional layers over the tree-structured skeleton joints, and perform convolution with triangular filters at each layer, as illustrated in Figure 6.

Table 2: Details of the hierarchy of dilated tree convolutions (corresponding to Figure 6).

Dilated tree convolutional layer index	#1	#2	#3
Dilation rate (d)	1	2	4
Perception sub-tree height	2	4	8
Output channels	75	75	75

Dilated convolutions which are effective and efficient in computation are also used here (similar to section 3.1). Only the top and the bottom nodes in each triangular region of each position are used for activation calculation, as shown by the Layer #1 (with dilation rate set to 1) and Layer #2 (with dilation rate set to 2) in Figure 6. Here we call this convolution design as dilated tree convolution.

Three dilated tree convolutional layers are stacked in our model, and their dilation rates are 1, 2, and 4, respectively. Therefore, a hierarchy of dilated tree convolutions are constructed over the skeletal data. The details of this hierarchy design are shown in Table 2.

With this design, the nodes in different layers of the hierarchy perceive different spatial ranges of the input skeleton joints. For example, *each node* in Layer #1 of Figure 6 learns a representation from a very local region of neighbouring joints of the input skeleton (perception sub-tree height is 2), while *each node* in Layer #2 learns a representation over a larger region of the skeleton (perception sub-tree height is 4). Specifically, the top layer, #3, can learn a representation based on all the joints of the whole skeleton tree (perception tree height is 8). This implies that the multi-level (local, mid-level, and holistic) structured semantic representations of the skeleton data are learned at different layers in this hierarchy.

Finally, we aggregate the multi-level representations by averaging the activations of all the convolutional nodes in the hierarchy, and the aggregated result is fed to our SSNet as the representation of the skeleton data at each frame (see Figure 2).

Since the multi-level structured semantic representations are learned, which are effective for representing the spatial structure and posture of the human skeleton at each frame, the performance of our SSNet for action prediction is improved. Moreover, this structured skeleton representation learning procedure can be attached to our SSNet as an input processing module of it (see Figure 2), such that the whole model of our SSNet is still end-to-end trainable.

3.6. Objective Function

The objective function of our SSNet is formulated as:

$$\ell = \ell_c(\hat{c}_t, c_t) + \gamma \ell_s(\hat{s}_t, s_t) \quad (5)$$

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 t. 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 our SSNet, we generate fixed-length clips from the annotated long sequences with sliding temporal windows. The length of each clip is equal to the perception temporal scale of the top convolutional layer (255 frames). Each clip can then be fed to the SSNet. In the training phase, class prediction is performed using the proper layer that is chosen based on the ground truth distance to the start point. We also observe adding small random noise to the layer choosing process during training is helpful for improving the generalization capability of our network for class prediction.

In the testing phase, the 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 distance regression result of its previous step. The ground truth information of the start point is not used during testing.

4. Experiments

The proposed method is evaluated on four challenging datasets: the OAD dataset [67], the ChaLearn Gesture dataset [73], the PKUMMD dataset [84], and the G3D dataset [49]. In all the 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 extensive experiments with the following different architectures:

1. **SSNet.** This is our proposed network for skeleton-based 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*. The multi-level structured skeleton representations are used in this network.

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 and propose 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. To make a fair comparison, skip connections (see Eq (3)) are also used in each FSNet, *i.e.*, all layers (corresponding to different scales) in a FSNet are connected as Eq (3) 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.*, exhaustive multi-scale multi-round scans are used to perform action prediction at each time step.

4. ***SSNet-GT*.** Beside the aforementioned models, we also evaluate 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.

Note that the multi-level structured skeleton representations are used in all of the above architectures (SSNet, FSNet ($\mathcal{S}$), FSNet-MultiNet, and SSNet-GT) for fair comparisons.

Our proposed approach is also compared to other state-of-the-art methods for skeleton-based activity analysis:

1. **ST-LSTM** [12]. This network achieves superior performance on 3D skeleton-based action recognition task. We adapt it to our online action prediction task and generate a prediction of the action class at each frame of the streaming sequence.

2. **JCR-RNN** [67]. 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.

3. **Attention Net** [85]. 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

The experiments are conducted with the Torch7 toolbox [86]. Our network is trained from scratch, *i.e.*, the network parameters are initialized with small random values (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. The output dimensions of FC1, FC3, FC4, and FC5 in Figure 3 are set to 50, 50, 50, and 1, respectively. FC2's output dimension is determined by the class number of each specific dataset. GLU [76] is the activation function used for the convolutional operations in our network (see Eq (2)). Residual connections [79] are used over different convolutional layers. The output channels of the convolutional nodes for temporal modeling (see Figure 2) are all 50. The output channels of the convolutional nodes for structured skeleton representation learning are equal to the dimension of the coordinate concatenation representation of a frame. In our experiment, γ in Eq (5) is set to 0.01. The above-mentioned parameters are obtained by cross-validation on the training sets.

In our SSNet, the proposed hierarchy of dilated tree convolution is used to learn the multi-level structured representation for each skeleton in a frame. If two skeletons are contained in a frame, then their structured representations are averaged. The averaged result is used as the representation for this frame.

We show the number of parameters of our SSNet with the two different skeleton representations in Table 3. By attaching the multi-level structured representation, the parameter number in the whole model of SSNet is only slightly larger than the configuration in which we use coordinate concatenation representation. This implies that the number of parameters in the hierarchy of dilated tree convolutions is quite small (only 13% of the whole model).

We also summarize the numbers of network parameters for different methods. The numbers of network parameters of SSNet, FSNet(15), FSNet(31), FSNet(63), FSNet(127), FSNet(255), FSNet-MultiNet, SSNet-GT, ST-LSTM, JCR-RNN, and AttentionNet are 310K, 170K, 200K, 240K, 270K, 310K, 1M, 310K, 420K, 290K, and 3M, respectively.

We perform our experiments with a single NVIDIA TitanX GPU. We evaluate the efficiency of our method for online action prediction in the streaming sequence, and show the running speed of it in Table 3. Our network responds fast for online action prediction. The low computational cost of our method is partially due to (1) the concise skeleton data as input, (2) the efficient dilated convolution, and (3) our activation sharing scheme. Besides, even if we learn the multi-level structured representations, the overall speed of our SSNet is still very fast.

4.2. Experiments on the OAD Dataset

The OAD dataset [67] was collected with Kinect v2 in daily-life indoor environments. Ten action classes were performed by different subjects. 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. In this dataset, 30 long sequences are used for training, and 20 long sequences are used for testing.

Table 3: Number of parameters and computational efficiency of our SSNet when using different skeleton representations within it.

Skeleton representations in SSNet	#Parameters	Speed
With coordinate concatenation	$270K$	$50fps$
With multi-level structured representation	$310K$	$40fps$

The action prediction results on the OAD dataset are shown in Figure 7 and Table 4. In the figures and tables, the prediction accuracy of an observation ratio $p\%$ denotes the average accuracy of the predictions in the observed segment ($p\%$) of the action instance.

Table 4: Action prediction accuracies on the OAD dataset. Note that in the last row, *SSNet-GT* is an "ideal" baseline, in which the *ground truth (GT)* scales are used for action prediction. Our SSNet, which performs prediction with the regressed scales, is even comparable to *SSNet-GT*. Refer to Figure 7 for more results.

Observation Ratio	10%	50%	90%
JCR-RNN	62.0%	77.3%	78.8%
ST-LSTM	60.0%	75.3%	77.5%
Attention Net	59.0%	75.8%	78.3%
FSNet (15)	58.5%	75.4%	75.9%
FSNet (31)	62.3%	75.2%	76.2%
FSNet (63)	62.2%	77.1%	78.9%
FSNet (127)	63.6%	76.3%	78.9%
FSNet (255)	57.2%	70.3%	71.2%
FSNet-MultiNet	62.6%	79.1%	81.6%
SSNet	**65.8%**	**81.3%**	**82.8%**
SSNet-GT	*66.7%*	*81.7%*	*83.0%*

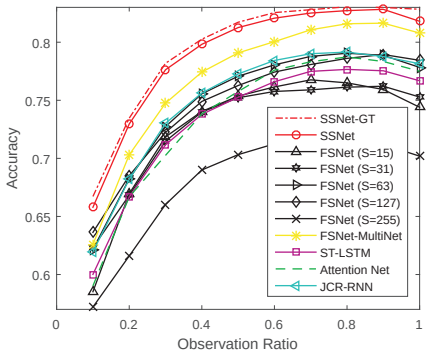


Figure 7: Action prediction results on the OAD dataset.

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 the current action at each step, is very effective for online action prediction.

The proposed SSNet significantly outperforms the state-of-the-art RNN/LSTM based methods, JCR-RNN [67] and ST-LSTM [12], which can handle continuous streaming skeleton sequences. The performance disparity could 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 [12, 67] 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. Our SSNet also outperforms the Attention Net [85] that assigns weights to differen frames and joints. This indicates the superiority of our SSNet with explicit scale selection.

We also observe the average action prediction accuracy decreases at the ending stages. A possible explantation is that the frames at the ending stages of some action instances contain postures and motions that are not very relevant to the current action's class label.

4.3. Experiments on the ChaLearn Gesture Dataset

The ChaLearn Gesture dataset [73] is a large-scale dataset for human action (body language) analysis, which consists of 23 hours of Kinect videos. A total of 20 action classes were performed by 27 subjects. This dataset is very challenging, as the body motions of many action classes are very similar.

Unlike the NTU RGB+D dataset [40], in which every video contains only one action, each video in the ChaLearn Gesture dataset includes multiple (8∼20) action instances. Thus this dataset is suitable for online action prediction. The starting and ending frames of 11116 action instances are annotated. On this dataset, $3/4$ of the annotated videos are used for training, and the remaining annotated videos are held for testing. We sample 1 frame from every 4 frames considering the large amount of data.

We report the action prediction results in Figure 8 and Table 5. Our SSNet outperforms other methods at all observation ratios on this large-scale dataset.

Table 5: Action prediction accuracies on the ChaLearn Gesture dataset. Refer to Figure 8 for more results.

Observation Ratio	10%	50%	90%
JCR-RNN	15.6%	51.6%	64.7%
ST-LSTM	15.8%	51.3%	65.1%
Attention Net	16.8%	52.1%	65.3%
FSNet (15)	16.6%	50.8%	62.0%
FSNet (31)	16.9%	53.2%	64.4%
FSNet (63)	15.8%	49.8%	60.8%
FSNet (127)	14.8%	46.4%	56.4%
FSNet (255)	14.5%	45.7%	55.4%
FSNet-MultiNet	17.5%	54.1%	65.9%
SSNet	**19.5%**	**56.2%**	**69.1%**
SSNet-GT	*20.1%*	*56.8%*	*70.0%*

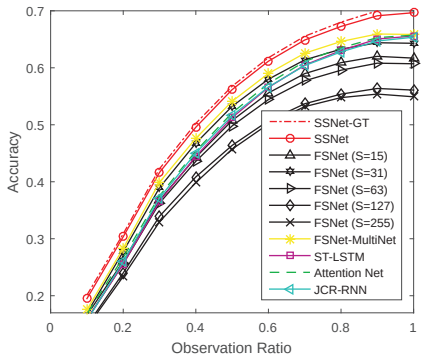


Figure 8: Action prediction results on the ChaLearn Gesutre dataset.

4.4. Experiments on the PKUMMD Dataset

The PKUMMD dataset [84] was captured for RGBD-based activity analysis in continuous sequences. Cross-subject evaluation protocol is used for this dataset, 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. The comparison results of

the prediction performance on this dataset are presented in Figure 9 and Table 6.

Our method achieves the best results at all the progress levels on this dataset. Specifically, our SSNet outperforms other methods significantly, even when only a very small ratio (10%) of the action is observed. This indicates that our method can produce a much better prediction at the early stage by focusing on the current action, compared to other methods which do not explicitly consider the scale selection.

Another observation is that the FSNet with fixed scale at each time step is quite 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.

Table 6: Action prediction accuracies on the PKUMMD dataset. Refer to Figure 9 for more results.

Observation Ratio	10%	50%	90%
JCR-RNN	25.3%	64.0%	73.4%
ST-LSTM	22.9%	63.0%	74.5%
Attention Net	19.8%	62.9%	74.9%
FSNet (15)	27.1%	67.4%	76.2%
FSNet (31)	30.6%	69.9%	79.8%
FSNet (63)	25.3%	63.5%	72.1%
FSNet (127)	25.9%	60.6%	71.0%
FSNet (255)	20.2%	50.9%	62.4%
FSNet-MultiNet	27.4%	71.8%	80.3%
SSNet	**33.9**%	**74.1**%	**82.9**%
SSNet-GT	*34.8%*	*74.2%*	*83.1%*

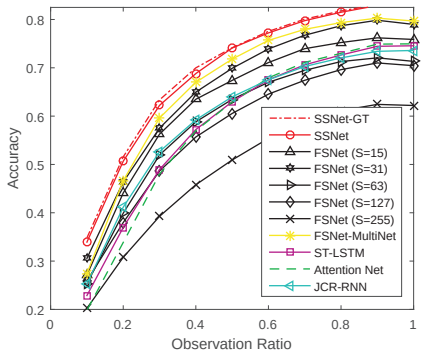


Figure 9: Action prediction results on the PKUMMD dataset.

4.5. Experiments on the G3D Dataset

The G3D dataset [49] containing 20 gaming actions was collected with a Kinect camera. There are 209 untrimmed long videos in this dataset. We use 104 videos for training, and the remaining ones are used for testing. Our SSNet achieves superior performance on this challenging dataset, as shown in Figure 10 and Table 7.

Table 7: Action prediction accuracies on the G3D dataset. Refer to Figure 10 for more results.

Observation Ratio	10%	50%	90%
JCR-RNN	70.0%	79.1%	81.9%
ST-LSTM	67.3%	75.6%	76.8%
Attention Net	67.4%	76.9%	79.3%
SSNet	**72.0**%	**81.2**%	**83.7**%
SSNet-GT	*73.5%*	*81.5%*	*84.0%*

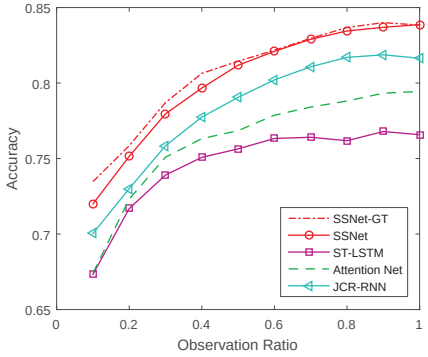


Figure 10: Action prediction results on the G3D dataset.

4.6. Evaluation of Skeleton Representations

We compare the performance of our SSNet when using the multi-level structured skeleton representations to that when using the coordinate concatenation representation, and report the results in Table 8.

The comparison results show that by using the hierarchy of dilated tree convolutions to learn the multi-level structured representation for the skeleton data in each frame, the action prediction performance of our SSNet is significantly improved. This clearly demonstrates the effectiveness of our newly proposed method in learning a discriminative representation of the human skeleton data in the spatial domain.

It is worth noting that, even if we do not use the powerful multi-level structured representation, but directly use the coordinate concatenation representation, our SSNet still outperforms the state-of-the-art skeleton-based activity analysis methods, JCR-RNN [67], ST-LSTM [12], and Attention Net [85], on all the four datasets.

Table 8: Action prediction accuracies (%) of SSNet with different skeleton representations.

Skeleton representations	OAD			ChaLearn Gesture			PKUMMD			G3D		
	Observation Ratio			Observation Ratio			Observation Ratio			Observation Ratio		
	10%	50%	90%	10%	50%	90%	10%	50%	90%	10%	50%	90%
Coordinate concatenation	65.6	79.2	81.6	17.5	53.5	65.9	30.0	68.5	78.6	70.1	79.1	82.0
Multi-level structured representation	**65.8**	**81.3**	**82.8**	**19.5**	**56.2**	**69.1**	**33.9**	**74.1**	**82.9**	**72.0**	**81.2**	**83.7**

Table 9: Start point regression performance ($SL\text{-}Score$). SSNet^C indicates that we use the coordinate concatenation representation for the network.

Dataset	JCR-RNN	SSNet^C	SSNet
OAD	0.42	0.69	**0.71**
ChaLearn Gesture	0.49	0.58	**0.60**
PKUMMD	0.61	0.72	**0.75**
G3D	0.62	0.72	**0.74**

4.7. Evaluation of Distance Regression

We adopt the metric $SL\text{-}Score$ proposed in [67] to evaluate the distance regression performance 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 the action instance. For false classification samples, the score is set to 0.

We report the regression performance of our SSNet in Table 9. As the action detection method, JCR-RNN [67], also estimates the start point, we also compare our method with it. Besides, we investigate the regression performance of the SSNet when we do not use multi-level structured representation but directly use coordinate concatenation for it (here we denote this case as SSNet^C).

The results show that our SSNet provides the best regression performance. Specifically, we observe that the regression result of SSNet (with multi-level structured representation) is better than SSNet^C (with coordinate concatenation). This indicates that by effectively learn the spatial tree structure of the input skeleton data, the accuracy of temporal distance regression can also be improved.

We also evaluate the average regression errors in the observed segment ($p\%$) on the large-scale ChaLearn Gesture dataset in Table 10. The regression error is calculated as $|\hat{s}-s|$. We find our method regresses the distance reliably. When only a small ratio (5%) of the action instance has been observed, the average regression error is 6 frames. The regression becomes more reliable when more frames are observed. We also visualize some examples in Figure 11. It shows that our SSNet achieves promising regression performance.

Table 10: Start point regression errors.

Observed Segment	5%	10%	30%	50%	70%	90%
Error (frames)	6	4	3	3	3	3

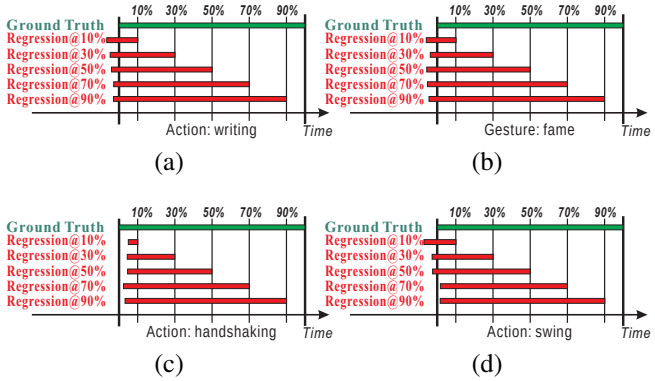


Figure 11: Examples of the start point regression results on the four datasets. The leftmost point of the green bar is the ground truth start point position of the current ongoing action. The leftmost point of each red bar (ending at $p\%$) is the regressed start point position when $p\%$ of the action instance is observed.

4.8. Evaluation of Network Configurations

We configure the maximum dilation rate 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 11 show that using more layers are beneficial for performance as the perception temporal window scale of the top layer increases. However, 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 activity analysis. Thus using the SSNet with 14 layers (with maximum window scale 255) is suitable.

We also evaluate the performance of our SSNet with different γ values (see Eq (5)) in Figure 12. We observe our SSNet yields the best performance when γ is set to 0.01.

As shown in Eq (3) and Eq (4), in the modules of generating representations for class prediction and distance regres-

Table 11: Evaluation of different configurations of the proposed network on the OAD dataset.

Number of 1-D convolutional layers	8	10	12	14	16
Maximum dilation rate	8	16	32	64	128
Maximum perception temporal window scale (frames)	*31*	*63*	*127*	*255*	*511*
Start point regression (SL-$Score$)	0.65	0.68	0.70	0.71	0.71
Prediction accuracy (%)	75.2	77.8	79.0	80.6	80.6

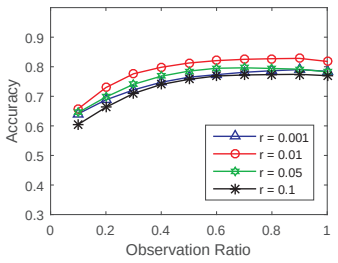


Figure 12: Action prediction results with different γ values on the OAD dataset.

Table 12: Frame-level classification accuracies. FSNet(best) denotes the FSNet that gives the best results among all FSNets.

Dataset	ST-LSTM	AttentionNet	JCR-RNN	FSNet(best)	SSNet
OAD	0.77	0.75	0.79	0.80	**0.82**
ChaLearn	0.62	0.63	0.62	0.64	**0.66**
PKUMMD	0.78	0.80	0.79	0.82	**0.85**
G3D	0.70	0.71	0.74	0.75	**0.76**

sion, instead of using the activation from one convolutional layer only, we add skip connections (links from the bottom convolutional layers). In our experiment, we observe that using this skip connection design, the action prediction accuracy can be improved by about 1.5%. We also investigate to further add batch normalization (BN) layers [87] to our network, and we do not see obvious performance improvement, thus BN layers are not used in our model.

4.9. Frame-level Classification Accuracies

As the action classification is performed at each frame of the videos, the average classification accuracies over all frames are also evaluated, and the results are reported in Table 12. The results show the superiority of our SSNet over the compared approaches.

5. Conclusion

In this paper, we have proposed a network model, SSNet, for online action prediction in untrimmed skeleton sequences. A stack of convolutional layers are introduced to model the dynamics and dependencies in temporal dimension. A scale selection scheme is also proposed for SSNet, with which our network can choose the proper layer corresponding to the most proper window scale for action prediction at each time step. Besides, a hierarchy of dilated tree convolutions are designed to learn the multi-level structured representations for the skeleton data in order to improve the performance of our network. Our proposed method yields superior performance on all the evaluated benchmark datasets. In this paper, the SSNet is proposed for handling the online action prediction problem. This network could also be extended to address the problem of temporal action detection in streaming skeleton sequences, which requires to locate each action in the skeleton sequence and meanwhile predict the class of each action. We leave this extension as our future work.

Acknowledgment

This research was carried out at Rapid-Rich Object Search (ROSE) Lab at Nanyang Technological University. ROSE Lab is supported by the National Research Foundation, Singapore, and the Infocomm Media Development Authority, Singapore. This work was supported in part by the National Basic Research Program of China under Grant 2015CB351806, and the National Natural Science Foundation of China under Grant 61661146005 and Grant U1611461. We acknowledge the NVIDIA AI Technology Centre (NVAITC) for the GPU donation.

References

[1] 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.

[2] 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.

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

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

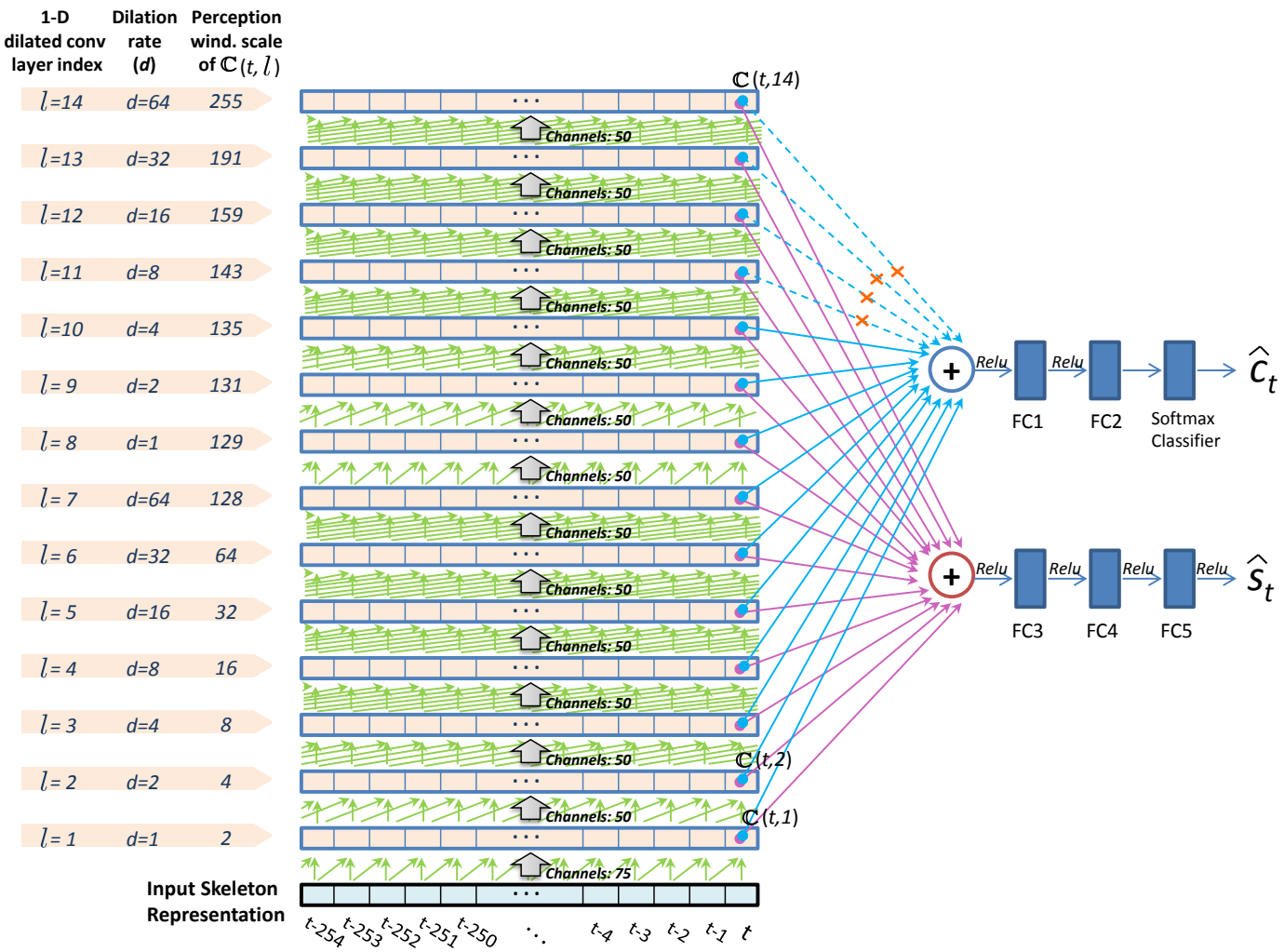


Figure 13: Detailed network architecture configurations of SSNet (for action prediction at the time step t). The distance regression is performed based on the top convolutional layer (together with the layers below it with the skip connections), which has a large perception window. The class prediction is performed based on the selected *proper* layer (together with the layers below it), which is selected based on the estimated window scale.

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

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

[7] 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.

[8] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *T-CYB*, 2013.

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

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

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

[12] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal lstm network with trust gates," *T-PAMI*, 2017.

[13] M. Liu, Q. He, and H. Liu, "Fusing shape and motion matrices for view invariant action recognition using 3d skeletons," in *ICIP*, 2017.

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

[15] 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," *T-IP*, 2018.

[16] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *ICCV*, 2017.

[17] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in *WACV*, 2014.

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

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

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

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

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

[23] P. Siva and T. Xiang, "Weakly supervised action detection." in *BMVC*, 2011.

[24] 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.

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

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

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

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

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

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

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

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

[33] 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 *CVPRW*, 2012.

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

[35] J. Hu, W. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for rgb-d activity recognition." *T-PAMI*, 2017.

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

[37] 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.

[38] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Computer vision for human-machine interaction," in *Computer Vision for Assistive Healthcare*, 2018.

[39] 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.

[40] 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.

[41] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, "Spatio-temporal graph convolution for skeleton based action recognition," *arXiv*, 2018.

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

[43] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *arXiv*, 2018.

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

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

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

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

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

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

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

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

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

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

[54] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Leveraging structural context models and ranking score fusion for human interaction prediction," *T-MM*, 2017.

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

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

[57] 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.

[58] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *ICCV*, 2017.

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

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

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

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

[63] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *CVPR*, 2017.

[64] J. Gao, Z. H. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," in *BMVC*, 2017.

[65] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017.

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

[67] Y. Li, C. Lan, J. Xing, W. Zeng *et al.*, "Online human action detection using joint classification-regression recurrent neural networks," in *ECCV*, 2016.

[68] 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.

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

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

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

[72] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "Ssnet: Scale selection network for online 3d action prediction," in *CVPR*, 2018.

[73] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *ICMI*, 2013.

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

[75] 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.

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

[77] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *T-PAMI*, 2017.

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

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

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

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

[82] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *CVPR*, 2017.

[83] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[84] C. Liu, Y. Hu *et al.*, "Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding," *arXiv*, 2017.

[85] 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.

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

[87] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.