Temporal Perceptive Network for Skeleton-Based Action Recognition

Yueyu Hu
huyy@pku.edu.cn
Chunhui Liu
liuchunhui@pku.edu.cn
Yanghao Li
lyttonhao@pku.edu.cn
Sijie Song
ssj940920@pku.edu.cn
Jiaying Liu*
liujiaying@pku.edu.cn

Institute of Computer Science and Technology
Peking University
Beijing, China

Abstract

The major challenge for skeleton-based action recognition is to distinguish the difference between various actions. Traditional Recurrent Neural Network (RNN) structure may lead to unsatisfactory results due to the inefficiency in capturing local temporal features, especially for large-scale datasets. To address this issue, we propose a novel Temporal Perceptive Network (TPNet) to enable the robust feature learning for action recognition. We design a temporal convolutional subnetwork, which can be embedded between RNN layers, to enhance automatical feature extraction for local temporal dynamics. Experiments show that the proposed method achieves superior performance to other methods and generates new state-of-the-art results. The model won the first place in the ACCV Workshop Large-Scale 3D Human Activity Analysis Challenge in Depth Videos.

1 Introduction

Human action recognition techniques facilitate a broad range of practical applications, *e.g.* video surveillance, video understanding, and human-computer interaction. It aims to classify the specific actions in a video according to its content. Different from image classification, the action recognition task involves temporal dynamics modeling, which makes the problem more challenging. To recognize human actions in a raw video clip, the features of motion and appearance should be considered jointly. Earliest works are first based on RGB video frames. It shows great potential in the field of action recognition for the methods based on densely sampled trajectories [26, 27] or Convolutional Neural Network (CNN) [5, 19, 28, 31].

Alongside with RGB based action recognition, there is a trend of exploiting devices

* Corresponding author.

This work was supported by National Natural Science Foundation of China under contract No. 61472011 and Microsoft Research Asia (project ID FY17-RES-THEME-013).

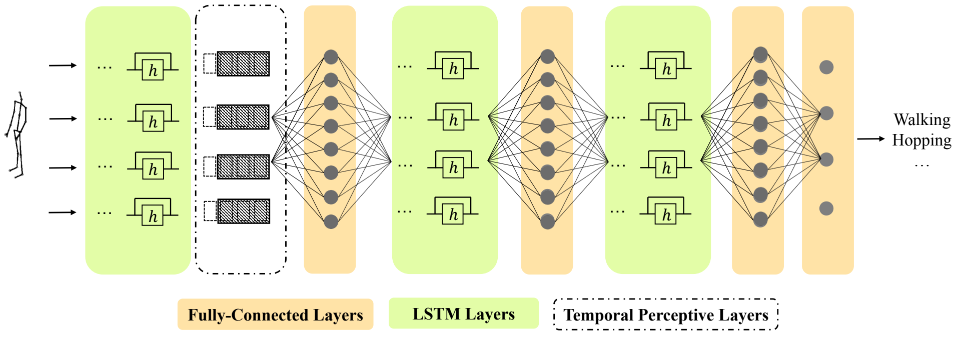


Figure 1: The architecture of the TPNet. The input raw skeleton sequence is made into a spatial-temporal tensor and pushed into the LSTM layer. One TP layer is placed after 1st LSTM layer for local temporal modeling. In the final LSTM layer, only the output of the last time step is sent to the final fully-connected layer for classification. Finally, the *SoftMax* activation predicts the probability for a sequence belonging to one action category.

equipped with color-depth sensing cameras like Microsoft Kinect to capture skeleton data for human action analysis. This kind of device can provide better observation of human body. According to the biological research [11], the skeleton data can provide the valuable and comprehensive representation of a series of human dynamics. Due to its unique characteristics, it shows great robustness to illumination variation, clustered background, and camera motion. Because skeleton data is of lower dimension compared with CNN features, to model its temporal dynamics can be much more time-efficient and power-efficient. By considering co-occurrences of local features [6, 17] and handcraft temporal representations [23, 30], semantic features for further classification can be extracted from skeleton representations. Some recent works have proposed new low-level graph based representations for classification [13, 29]. And in deep learning method, due to succinctness of skeleton data, recurrent neural network (RNN) models can be naturally used to model temporal dynamics [14, 32]. Applying RNN derived networks like Long Short-Term Memory (LSTM) [9] or Gated Recurrent Units (GRU) [1] is one way to achieve better performance for skeleton based action recognition. Though the LSTM network is designed to capture both long-term and short-term feature for temporal-distributed sequences, its ability to model some extra short-term dynamics is restricted. Once the video clips go longer, this drawback significantly affects the improvement of recognition performance.

In this paper, we propose a novel Temporal Perceptive (TP) subnetwork. Our motivation comes from the rethinking of how we view an object in the real world. To fully understand one object, we need to both view it as a whole system as well as view it in detail to investigate its mechanism. This kind of idea is also employed in the field of designing and analyzing CNNs with the concept of receptive field [8, 16].

Figure 1 shows our end-to-end model for skeleton action recognition. The sequential feature map generated by the 1st LSTM layer is pushed into the Temporal Perceptive subnetwork with temporal convolution. The extracted local feature maps are then fed into the following feed-forward layer. In the final LSTM layer, only the output of the last time step is sent to the final fully-connected layer for classification. Finally, the *SoftMax* activation predicts the probability for a sequence belonging to one action category. With a joint training of the whole network, the parameters of the convolution kernel are tuned to be able to model fine-grained temporal features. The full model is able to capture both long-term semantic action feature and extra-short-term motion dynamics. Experimental results show that our proposed TP layer can further improve the accuracy of action recognition. By visualizing the weights of the convolutional kernel, we are convinced that this layer learns the optimized transformation we expect the network to learn during joint training.

2 Related Works

In this section, we review previous works related to our work on *deep learning for action recognition* and *local temporal modeling*.

Deep learning for action recognition: Many deep learning based methods have been proposed for action recognition on both raw RGB modality and skeleton modality. For RGB frames, it is common to separately train a temporal model and a spatial model to jointly model temporal dynamics and appearance feature [5, 19, 28]. Optical flow maps for frame sequence are extracted and stacked before they are fed into the sophisticated CNNs like ResNet [8] or GoogleNet [22]. The CNN models are first fine-tuned for the target dataset to reduce cross-dataset variation using frames in the video clips. With a network fusion of the temporal and spatial subnetwork, the prediction result is given through the final *SoftMax* layer. These methodologies achieved promising performance on large-scale action recognition benchmarks like UCF101 action datasets[21].

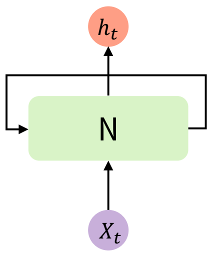
Local temporal modeling: With the development of color-depth sensing devices, methods for high-performance action recognition on human skeleton data have been proposed. Action recognition problems can be approached by simply applying LSTM on raw skeleton data. In [32], bidirectional LSTM is exploited to model long-term and short-term feature of skeleton dynamics. In order to enable the network to learn to model the motion of different parts of the body separately, full-connected layers with weights regularizations are added to the network to automatically learn the co-occurrence of the joints. Song *et al.* [20] proposed the spatial and temporal attention mechanism to capture the salient part of the human skeleton and motion sequence to further reduce noise and enhance feature capturing. In the work of [30], they calculate the acceleration and velocity of each joint and treat them as the local feature for further classification. This non-parametric approach is artificially defined and may not be sufficient to accurately model local dynamics. Veeriah *et al.* [25] proposed a modified LSTM unit to learn salient dynamic patterns with high order derivatives during the training of LSTM using Back Propagation Through Time. However, methods discussed above either have such heavy parameter space that training and inference are computation consuming and are easily leading to an unsatisfactory local minimum or are weak at exploring local temporal features. In [3], a method based on deep CNN is proposed for skeleton action recognition, which is a totally different approach using CNN for spatial-temporal modeling exploiting the sophisticated deep convolution. However, it is based on CNN and sequences have to be downsampled or cropped to the same length to fit the input size. And it can hardly be used to form a real-time system.

Methods mentioned above mainly focus on the long-term dynamics, while local temporal modeling is also important to recognize actions involving subtle motions. Thus, we propose the TP layer to model extra-short-term dynamics without introducing many parameters. Our proposed method exploits a general temporal convolutional mechanism based on LSTM structure to learn the local feature of skeleton sequence. As the convolution is only applied on the temporal dimension with few additional parameters, the computation is efficient. The learned parameters of the convolutional layer enhance the learning ability of the whole network, compared with non-parametric solutions. The proposed subnetwork is compatible with both raw skeleton sequence and time-distributed high-level LSTM feature, which means its ability to enhance local feature extraction is universal.

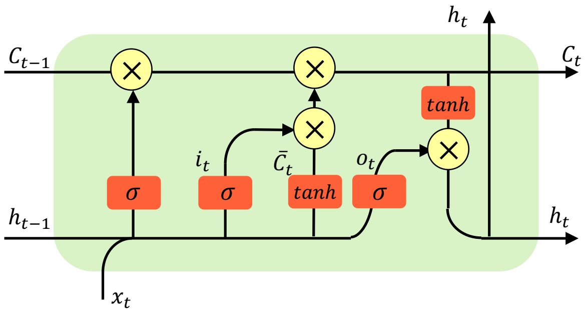
3 Overview of RNN and LSTM

In this section, we briefly review the Long Short-Term Memory (LSTM) network, which is one particular kind of Recurrent Neural Network (RNN) widely used for temporal dynamics modeling.

Figure 2(a) shows the basic structure of RNN. The RNN is a kind of neural networks where connections between units form a directed cycle. The structure enables the neuron to generate a new output according to both its history and the new input. This feature enables the temporal dynamics modeling. However, the learning capability of RNN in the temporal



(a)



(b)

Figure 2: Structure of RNN and LSTM. (a) Simple RNN, (b) LSTM Unit.

field is largely restricted by gradient vanishing effect [9]. It fails to model some long-term dynamics necessary for a long video clip. So LSTM is developed as an improved RNN to address this problem [7, 9], as shown in Figure 2(b). LSTM is able to maintain and process information over time with a self-connected memory cell c_t. At each time step, the network can choose to read, write or reset the memory cell governed by the input gate i_t, forget gate f_t and output gate o_t. Their activations can be summarized as follows,

$$\begin{aligned}
\mathrm{i_t} &= \sigma(\mathbf{W}_{xi}\mathrm{x_t} + \mathbf{W}_{ht}\mathrm{h_{t-1}} + \mathbf{W}_{ci}\mathrm{c_{t-1}} + \mathrm{b_i}), \\
\mathrm{f_t} &= \sigma(\mathbf{W}_{xf}\mathrm{x_t} + \mathbf{W}_{hf}\mathrm{h_t} - 1 + \mathbf{W}_{cf}\mathrm{c_{t-1}} + \mathrm{b_f}), \\
\mathrm{c_t} &= \mathrm{f_t} \odot \mathrm{c_{t-1}} + \mathrm{i_t} \odot \tanh(\mathbf{W}_{xc}\mathrm{x_t} + \mathbf{W}_{hc}\mathrm{h_{t-1}} + \mathrm{b_c}), \\
\mathrm{o_t} &= \sigma(\mathbf{W}_{xo}\mathrm{x_t} + \mathbf{W}_{ho}\mathrm{h_{t-1}} + \mathbf{W}_{co}\mathrm{c_t} + \mathrm{b_o}), \\
\mathrm{h_t} &= \mathrm{o_t} \odot \tanh(\mathrm{c_t}).
\end{aligned} \tag{1}$$

Here $\odot$ denotes the element-wise product, $\sigma(x)$ denotes the sigmoid function. By applying these operations we have an output response h_t for each neuron at every time step. With the help of these inner gates, LSTM units can now learn to *remember* and *forget*, acting more like a biological neuron of remembering. This kind of structure can take care of both long-term and short-term information, making it practical for temporal modeling. At the final LSTM layer, we select the last $\mathbf{h_t}$ as it contains feature of all the time steps.

4 Proposed Method

In this section, we introduce our end-to-end model for action recognition with temporal perceptive layer. We design the model to automatically learn a temporal convolutional kernel for local feature representation.

4.1 Motivation

Our motivation comes from the data augmentation techniques and the common practice of employment of the velocity and the acceleration in skeleton-based action recognition. Data augmentation can effectively boost feature learning and improve the classification performance especially for small-scale datasets. To adapt to large-scale datasets which already contain enough variation in the training samples, it is necessary to enable parametric learning to selectively augment certain aspect of the data. Following this idea, we employ convolutional operations in RNN-based deep neural networks. By producing several feature maps from one sequence through different filter, different aspect of the feature are evaluated and selected. Traditional skeleton-based action recognition methods use velocity and acceleration information as hand-crafted features for further classification [30]. It is an effective practice for cases like discriminating between punching and pushing. In discrete form, velocity v_t and acceleration a_t at time t can be expressed as,

$$\begin{aligned} v_t &= p_t - p_{t-1} = [1,-1] \odot [c_t, c_{t-1}], \\ a_t &= v_t - v_{t-1} = [1,-2,1] \odot [c_t, c_{t-1}, c_{t-2}], \end{aligned} \tag{2}$$

where c_t denotes the coordinate of a joint at the t^{th} time step and $\odot$ is the element-wise multiplation. We argue that there may exist other forms of feature which can be expressed as a linear transformation of the incoming data or feature map, just like the velocity and acceleration mentioned above. This form of features only account for local data and is extracted using convolutional operations, as,

$$\mathbf{o_t} = K_{[1,N]} \odot H_{[t+1,t+N]}, \tag{3}$$

where $K_{[1,N]}$ is the temporal convolution kernel of length N and $H_{[t,t+N]}$ stands for a part of the input sequence along temporal axis.

4.2 Temporal Perceptive Subnetwork

The Temporal Preceptive layer is one auto-adaptive component of the LSTM-based network for temporal modeling. It handles time-distributed sequential data and do parametric linear transformation to the data. For the incoming feature map $W_{T,H}$ which stands for a sequence of T time steps and at each time step the feature vector is of length H. We treat the feature

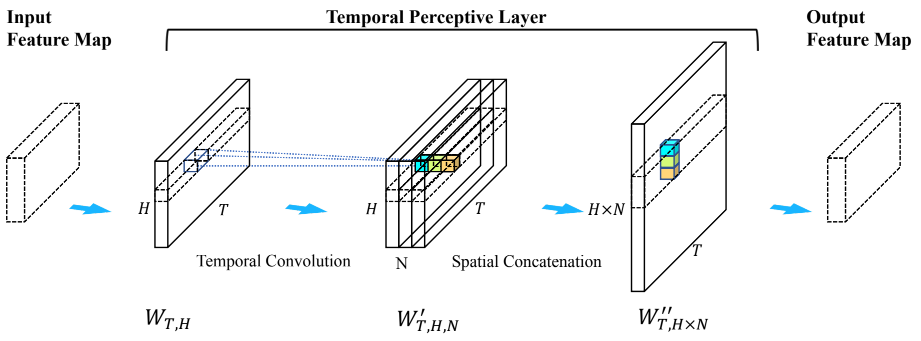


Figure 3: The Structure of The Temporal Perceptive Network. Given incoming skeleton map or feature map $W_{H,T}$ representing sequence of T time steps and H spatial dimensions, we first conduct a temporal convolution of N filters to make $W'_{T,H,N}$. $W'_{T,H,N}$ is then concatenated along the last axis, expanding the spatial dimension to $H \times N$.

map as one channel. After its convolutional operation againest N filters of kernel $K_{m,n}$, the output feature map is $W'_{T,H,N}$ as there are N generated channels of N filters. These N channels are concantenated to form a feature vector of length $N \times H$. So the output of this layer is $W''_{T,N\times H}$. Here we state that the feature map can also be raw incoming skeleton data.

The architecture of temporal perceptive layer is shown in Figure 3. With the input data flow, the subnetwork is able to extract the short-term statistics automatically in temporal field. When doing the feed-forward process of the training, the convolution kernel will conduct a convolution operation to several time-steps. For a given sequence $[\dots, x_{i-1}, x_i, x_{i+1}, \dots]$, we have a kernel $[k_1, k_2, \dots, k_N]$ of length N. The output y_i of the layer l can be calculated as,

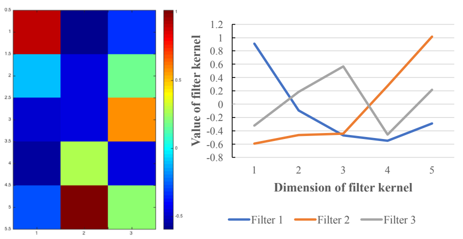
$$y_i^{(l)} = \sum_{j=1}^{N} \left(x_{i-N+j}^{(l)} \cdot k_j^{(l)} \right). \tag{4}$$

In the back-propagation phrase, for each temporal convolutional layer in TP subnetwork, the kernel weights k^l is updated with the derivatives,

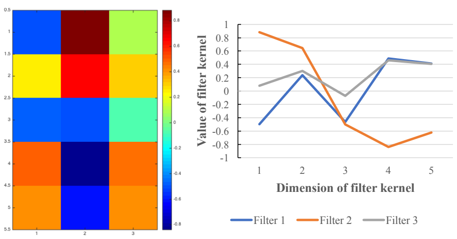
$$\frac{\partial L}{\partial k_t^{(l)}} = \sum_{i=1}^{T} \delta_i^{(l+1)} x_{i-N+t}^{(l)}, \tag{5}$$

where L denotes the loss value in one batch of the training, t is to iterate along each dimention of the kernel weights, and $\delta^{(l)}$ stands for the error of the l^{th} layer.

We visualize the learned weights of the kernels of the TP layer, shown in Figure 4. As we initialize the weights with random values, the model is proved to have learned its weights during training. The two sets of filters are placed in different places in the model. We can see that each filter learns different transformation for the input data, but both Figure 4(a) and Figure 4(b) have at least one filter with descending or increasing weight distribution (in the middle of the color map). It shows that our model learns to extract the derivatives of joint coordinates, which is similar to velocity. But the auto-extracted features are more general and discriminative. It adds to the robustness and diversity of feature learning to the whole model. We prove that our novel temporal perceptive mechanism can capture extra short-term feature to enhance the feature learning of LSTM.



(a)



(b)

Figure 4: Visualization of learned weights of the TP layers. Figure 4(a) is the learned weights when the TP layer is placed after the first LSTM layer, and Figure 4(b) when the TP layer is placed as the preprocessing unit. The color maps on the left visualize the actual weights and the graphs on the right show the distribution of the learned weights.

4.3 End-to-End Model Structure

Figure 1 shows the overall structure of our proposed model for skeleton action recognition. The LSTM is for sequential learning, especially for long-term modeling. The first two LSTM layers feed the whole sequence of the hidden states as the output response to the next layer. The output response contains temporal information which can then be detected and transformed using our novel temporal perceptive layer.

In our temporal perceptive network, convolutional operations are exerted on the sequential incoming data to explore temporal correlation. Different from previous LSTM layers, neurons in TP layers are aware of subtle dynamics of the sequence as one neuron has limited connections. This layer has the ability to adapt to the actual distribution of the incoming data so this kind of local exploration are for both raw joints and processed features from the previous layer.

To assist temporal modeling, feed-forward layers, where one neuron is connected to all neurons of the previous layer through trainable weights, are designed to model spatial structure. The weights are tuned during training to be aware of the co-occurrence of spatially distributed joints or output responses of the previous layer.

The model is implemented using Keras [2] and Theano [24]. We train the network using the Adam [12]. To deal with sequences in different length, masking is applied to the networks. The feed-forward layers and TP layers propagate the mask through the whole computing graph. A *SoftMax* activation is placed at the final layer to support computation of categorical cross-entropy loss.

5 Experimental Evaluation and Result

5.1 Datasets

5.1.1 NTU RGB+D Dataset (NTU)

The NTU dataset [18] is a large-scale dataset for action recognition. It contains as many as 56880 video clips in 60 categories, taken by three Microsoft Kinect sensors from different

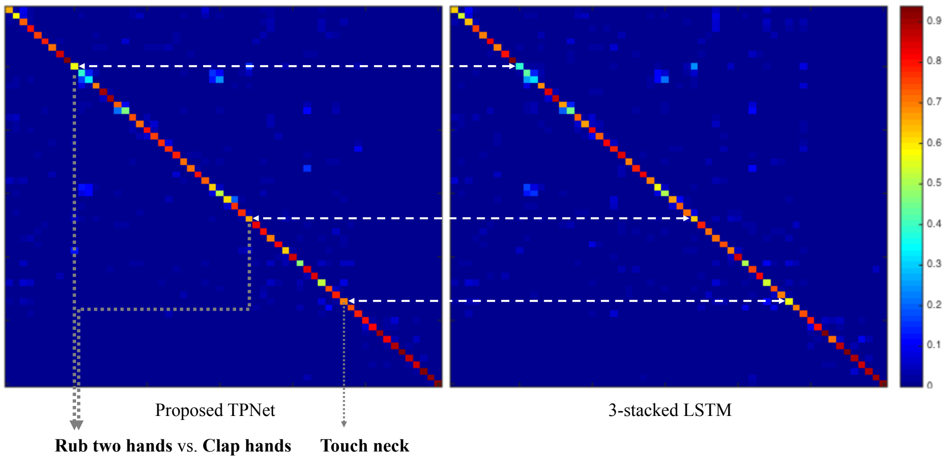


Figure 5: Confusion matrix of our original 3-stacked model and our proposed TPNet. Detailed description of the 60 action classes can be found at the website of the NTU RGB+D Action Recognition Dataset .

views in the indoor environment with 40 participants. Each category has an aggregated action class. The variation of subjects and view requests for robust modeling. Here we used only the skeleton data for training and testing. Note that we do the view normalization on the data [18]. We evaluate our algorithm on the settings of cross-subject and cross-view, respectively, which stands for a different view and different participants in the data collection process.

5.1.2 Kinect Interaction Dataset (SBU)

The SBU dataset is an interaction action recognition dataset with two subjects. It contains 230 sequences of 8 classes (6614 frames). We do the common subject independent 5-fold cross validation. We smooth each joint's position of the skeleton in the temporal domain to reduce noise [4, 32].

5.1.3 CMU Action Dataset

We used the categorized CMU motion capture dataset, mentioned in [32]. The categorized dataset contains 2,235 sequences of skeleton data. This dataset features a large variation of the length of the sequence, from 22 to 2378. We evaluate our proposed model using the 3-fold cross validation on the full dataset.

5.2 Ablation Analysis

Table 1 shows our experiments of the cross-subject validation on NTU dataset. For baseline model, we use the 3-stacked LSTM network with fully connected feed-forward layers in between. Number of neurons for the LSTM layers are 100×2, 110×2 and 200×2 where $\times2$ means the bidirectional setting. The number of neurons for feed-forward layers are 100 and 110, corresponding to those of the LSTM layers. This baseline model achieves promising basic results, and our proposed method improves the accuracy. For *TPNet input* setting,

Table 1: Different settings in our experiments for cross-subject validation on NTU dataset. For *TPNet input* setting, we place the TP layer right after the input of the model. For *TPNet feature* setting, the TP layer is placed right after the first bidirectional LSTM layer. Our novel temporal perceptive later can capture extra short-term feature to enhance the feature learning of LSTM and improves accuracy.

Model	Acc. (%)
Deep LSTM [18]	60.70%
STA-LSTM [20]	73.40%
Bidirectional LSTM	74.04%
TPNet input	74.86%
TPNet feature	**75.33%**

Table 2: Comparisons on the NTU dataset in accuracy.

Model	Acc. (%)(CS)	Acc.(%)(CV)
Dynamic Skeletons [10]	60.2%	65.2%
HBRNN [4]	59.1%	64.0%
Part-aware LSTM [18]	62.9%	70.3%
Deep LSTM [18]	60.7%	67.3%
ST-LSTM (Tree Traversal) + Trust Gate [15]	69.2%	77.7%
STA-LSTM [20]	73.4%	81.2%
Our Proposed Model	**75.3%**	**84.0%**

we place the TP layer right after the input of the model. It is to treat the TP layer as the preprocessing element of the whole network. We set the length of the temporal convolution kernel to be 5, which is a moderate length for the incoming frame rate. After the processing, the original dimension of 150 for the input data will be expanded to 450 as we set up 3 filters for the convolution. For *TPNet feature* setting, the TP layer is placed right after the first bidirectional LSTM layer which produces the whole sequence of the time-distributed feature. The output dimension of the first LSTM layer is 200 and it is expanded to 600 with 3 filters. As we can see from the results, temporal modeling on slightly more abstract feature performs better than the preprocessing-style temporal modeling.

To further examine what leads to the improvement of accuracy, we visualize the confusion matrix of our original 3-stacked model and our proposed TPNet respectively. As is shown in Figure 5, the proposed model can better discriminate actions with subtle extra-short-term, like *rubbing two hands* and *clapping hands*. Note that we follow the definition of action classes in the descriptions of the NTU RGB+D dataset[1] . With the proposed temporal convolution, our TPNet can better model short-term temporal dynamics without introducing much parameters and bring performance improvement.

5.3 Comparison with State-of-the-Art

We compare several existing methods for skeleton-based action recognition on the three datasets. On SBU and CMU datasets, whose scale is relatively small, we set the initial learning rate of Adam optimizer to 0.0001 and we set a batch size of 8 and 32. For SBU

[1]https://github.com/shahroudy/NTURGB-D

Table 3: Comparisons on the SBU dataset in accuracy.

Model	Acc. (%)
HBRNN-L [4]	80.4 %
Co-occurance RNN [32]	90.4 %
STA-LSTM [20]	91.5 %
ST-LSTM (Tree Traversal) + Trust Gate [15]	93.3 %
Our Proposed Model	**100.0%**

Table 4: Comparisons on the CMU dataset in accuracy.

Model	Acc. (%)
HBRNN-L [4]	75.02 %
Co-occurance RNN [32]	81.04 %
Our Proposed Model	**99.47%**

and CMU dataset, our proposed method reaches extraordinary high accuracy, proving the robustness of our model. On NTU dataset, we evaluate our model on both cross-subject and cross-view validations. We set the initial learning rate of Adam optimizer to 0.01 and use a batch size of 256. We achieve high accuracy of predition on both testing settings.

6 Conclusion

In this paper, we discuss our proposed model for automatically extracting the local temporal feature from skeleton data to improve the accuracy in large-scale action recognition. By introducing the novel TPNet structure, the model achieves state-of-the-art performance in large-scale action recognition challenge. Experimental results and visualization show that the convolutional kernels are tuned to extract local temporal statistics to enhance feature extraction of existing LSTM models.

References

[1] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv*, 2014.

[2] François Chollet. Keras. https://github.com/fchollet/keras, 2015.

[3] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *Proc. IAPR Asian Conference on Pattern Recognition*, pages 579–583. IEEE, 2015.

[4] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016.

[6] Yusuke Goutsu, Wataru Takano, and Yoshihiko Nakamura. Motion recognition employing multiple kernel learning of fisher vectors using local skeleton features. In *Proc. IEEE Int'l Conf. Computer Vision Workshops*, pages 79–86, 2015.

[7] Alex Graves. Supervised sequence labelling with recurrent neural networks, 2012.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 5344–5352, 2015.

[11] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. International Conference for Learning Representations*, 2015.

[13] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. *Tensor Representations via Kernel Linearization for Action Recognition from 3D Skeletons*, pages 37–53. 2016.

[14] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *Proc. European Conference on Computer Vision*, 2016.

[15] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Proc. European Conference on Computer Vision*, pages 816–833, 2016.

[16] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems*, pages 4898–4906, 2016.

[17] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Alberto Bimbo, and Pietro Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, pages 479–485, 2013.

[18] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016.

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[20] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proc. AAAI Conf. on Artificial Intelligence*, 2017.

[21] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, 2012.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2015.

[23] Lingling Tao and René Vidal. Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision Workshops*, pages 61–69, 2015.

[24] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv*.

[25] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision*, 2015.

[26] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 3551–3558, 2013.

[27] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 3169–3176, 2011.

[28] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *Proc. European Conference on Computer Vision*, pages 20–36, 2016.

[29] Pei Wang, Chunfeng Yuan, Weiming Hu, Bing Li, and Yanning Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *Proc. European Conference on Computer Vision*, pages 370–385, 2016.

[30] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2752–2759, 2013.

[31] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016.

[32] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 3697–3704, 2016.