DMM-Pyramid Based Deep Architectures for Action Recognition with Depth Cameras

Rui Yang[1,2] and Ruoyu Yang[1,2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] Department of Computer Science and Technology, Nanjing University, China
ryang@smail.nju.edu.cn, yangry@nju.edu.cn

Abstract. We propose a method for training deep convolutional neural networks (CNNs) to recognize the human actions captured by depth cameras. The depth maps and 3D positions of skeleton joints tracked by depth camera like Kinect sensors open up new possibilities of dealing with recognition task. Current methods mostly build classifiers based on complex features computed from the depth data. As a deep model, convolutional neural networks usually utilize the raw inputs (occasionally with simple preprocessing) to achieve classification results. In this paper, we train both traditional 2D CNN and novel 3D CNN for our recognition task. On the basis of Depth Motion Map (DMM), we propose the DMM-Pyramid architecture, which can partially keep the temporal ordinal information lost in DMM, to preprocess the depth sequences so that the video inputs can be accepted by both 2D and 3D CNN models. The combination of networks with different depth is used to improve the training efficiency and all the convolutional operations and parameters updating are based on the efficient GPU implementation. The experimental results applied to some widely used benchmark outperform the state of the art methods.

1 Introduction

Human action recognition is an important topic in computer vision. As a key step in an overall human action understanding system, action recognition is applied for many applications including human computer interaction, video surveillance, game control system and etc [1, 2]. Based on the traditional video sequences captured by RGB cameras, spatio-temporal features are widely used for the recognition task [3]. With the recent development of high-speed depth cameras, we can capture depth information of body's 3D position and motion in real-time [4]. Compared with the 2D information, the depth information has obvious advantages because it can distinguish the human actions from more than one view because the z-index displacement information which is lost in 2D frames is valued here. The new data format has motivated researchers to propose more innovative methods which are able to make full use of the depth information.

Convolutional neural network [5, 6] is an efficient recognition algorithm which is widely used in pattern recognition, image processing and other fields. The weight sharing network structure is of significance to reduce the complexity of network model and more similar to the biological neural network. As a special design of a multi-layer perceptron for the recognition of 2D shapes, this network structure has high invariance to translation, scaling, inclination and some other anamorphosis. [6–9] have shown CNNs' powerful ability in visual object recognition on the premise of appropriate training and parameter adjustment. In the field of human action recognition, [10] treats video frames as still images and apply CNNs to recognize actions at the individual frame level. [11–13] successfully extract spatial and the temporal features by performing 3D convolutions. But, impressive as these successes are, few research has been done on action recognition with the depth inputs.

In this paper, we firstly propose a 2D-CNN based deep model for human action recognition using depth maps captured by Kinect. The challenging datasets in this field usually provide us a lot of video clips and each clip only perform one complete action. Since traditional CNN is good at dealing with 2D inputs (usually the natural images), we need to engrave the frame sequences along the time axis into a static image before building the neural network model. The overall shape and position after superposition are used to indicate the action performed by the clip. Motion History Image (MHI) [14, 15] and the Motion Energy Image (MEI) [15] are two great engraving methods due to their simplicity and good performance. Here we use Depth Motion Maps (DMM) [16] which looks like MHI to a certain extent. DMM can accumulate global activities through entire video sequences to represent the motion intensity but the temporal ordinal relationship is lost. So we extend the DMM to DMM-Pyramid in order to avoid losing too much temporal features. We regard all the DMMs in a DMM-Pyramid as different channels of an image and the image is the final input for our architecture. With the steps of DMM-Pyramid calculation done, a modified CNN model is built to achieve the recognition result.

Secondly we propose a 3D-CNN based deep model in order to learn spatio-temporal features automatically. Compared with the preprocessing work in 2D-CNN model, the most difference is that here we divide the depth sequence in a clip evenly into N parts and apply DMM calculation to these parts respectively (In fact, it can be seen as the bottom layer of DMM-Pyramid). Then we stack multiple contiguous DMMs together to form a DMM cube (with size $Width \times Height \times N$) rather than a group of individual DMMs in 2D architecture. After that we convolute a 3D kernel to the cube. The remanning work is quite similar to the previous model. Both of the 2D/3D models are evaluated on two benchmark datasets: MSR Action3D dataset [1] and MSR Gesture3D dataset [17] which are are captured with depth cameras.

The key contributions of this work can be summarized as follows:

- We propose to apply 2D/3D convolutional networks to recognize the human actions captured by depth cameras. We use convolution operation to extract spatial and temporal features from low-level video data automatically.
- We extend DMM to DMM-Pyramid and then we can organize the raw depth sequence into formats which can be accepted by both 2D and 3D convolu-

tional networks. The preprocessing work is simple enough and will keep the raw information as much as possible.

- We propose to combine deep CNN models with different depth to further boost the performance. We train multiple models at the same time and apply a linear weighted combination to their outputs. Experimental result has proved the operation's effectiveness.
- We evaluate our models on the MSR Action3D dataset and MSR Gesture3D dataset in comparison with the state-of-the-art methods. Experimental results show that the proposed models significantly outperforms other ones.

The rest of the paper is organized as follows. Section 2 reviews the recent research work on human action recognition using the advantages of depth data. Section 3 and section 4 describe the 2D/3D-CNN architectures we proposed in detail. The experimental results and comparisons are given in section 5. Section 6 concludes the paper with future work.

2 Related works

Li et al [1] model the dynamics of the action by building an action graph and describe the salient postures by a bag-of-points (BOPs). It's an effective method which is similar to some traditional 2D silhouette-based action recognition methods. The method does not perform well in the cross subject test due to some significant variations in different subjects from MSR Action3D dataset.

Wu et al [18] extract features from depth maps based on Extended-Motion-History-Image (Extended-MHI) and use the Multi-view Spectral Embedding (MSE) algorithm. They try to find the frames that are similar to the beginning and ending frame in the unsegmented testing video sequence for temporal segmentation.

Yang et al [16] are motivated by the success of Histograms of Oriented Gradients (HOG) in human detection. They extract Multi-perspective HOG descriptors from DMM as representations of human actions. They also illustrate how many frames are sufficient to build DMM-HOG representation and give satisfactory experimental results on MSR Action3D dataset. Before that, they have proposed an EigenJoints-based action recognition system by using a NBNN classifier [19] with the same goal.

In order to deal with the problems of noise and occlusion in depth maps, Jiang et al extracts semi-local features called random occupancy pattern (ROP) [20]. They propose a weighted sampling algorithm to reduce the computational cost and claim that their method performs better in accuracy and computationally efficiency than SVM trained by raw data. After that they further propose Local Occupancy Patterns (LOP) features [21] which are similar to ROP in some case and improve their results to some extent.

Oreifej et al [22] propose to capture the observed changing structure using a histogram of oriented 4D surface normals (HON4D). They demonstrate that their method captures the complex and articulated structure and motion within the sequence using a richer and more discriminative descriptor than other ones.

Different from all above approaches, our methods do not try to extract any complex or so-called "rich" features from the depth sequences. We just leave the task of building high-level features from low-level ones to the deep CNN models. The preprocessing work on the raw inputs is quite simple.

3 A 2D-CNN Architecture

This section describes the 2D-CNN based deep model in detail. It shows how we stack contiguous frames together into still images and convert action recognition to a task which looks like traditional image classification based on CNN.

3.1 Depth Motion Map

[16] details the framework to compute action representation of DMM-HOG. In this subsection, the HOG descriptor is no longer needed. The only thing we care about is using DMM to stack depth sequences into the inputs which can be accepted by CNN model.

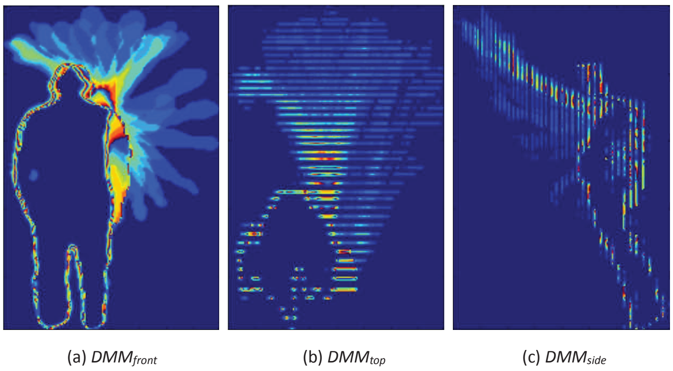
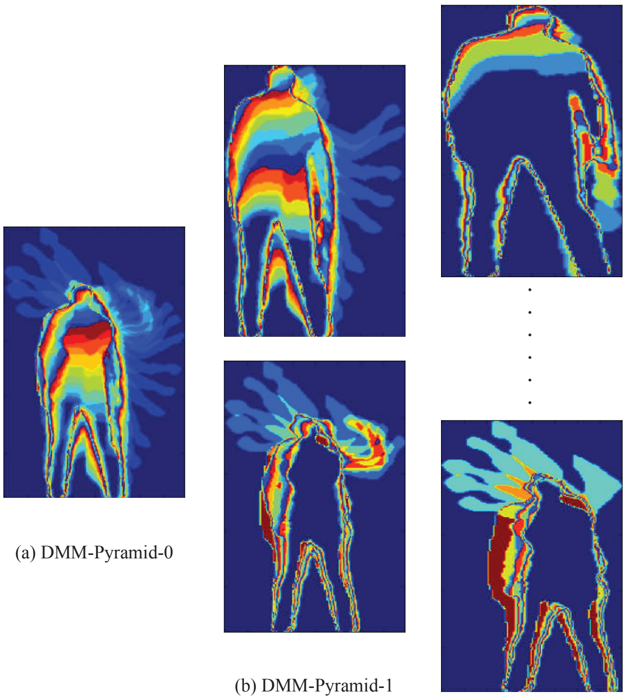


Fig. 1. Three DMMs represent the action "high arm wave". They summarize the body motion in the depth sequence from three orthogonal views.

To put it simply, DMMs are used to summarize the difference between each two consecutive depth maps in a clip. Each 3D depth frame generates three 2D binary maps including front views map map_f, side views map map_s, and top views map map_t. Then DMM is denoted as:

$$DMM_v = \sum_{i=1}^{N-1} |map_v^i - map_v^{i+1}|, \tag{1}$$

where $v \in \{front, side, top\}$ and N is the number of frames in a given clip. Figure 1 shows an example of three DMM maps generated by the depth maps sequence which performs the action "high arm wave".



(a) DMM-Pyramid-0

(b) DMM-Pyramid-1

(c) DMM-Pyramid-2

Fig. 2. A DMM-Pyramid architecture represents the action “pick up & throw” in different levels.

3.2 DMM-Pyramid

While we utilize DMM to summarize each whole video clip, we can not avoid losing some temporal information. For an example, there is a complex action called “pick up & throw” in the benchmark. This action contains two consecutive sub-actions: “bend to pick up” and “high throw”. If we only use one “general” DMM (even from three perspectives) to do the representation work, the confusion may occur between “pick up & throw” and “high throw” or “high arm wave” because their total depth motions are similar to each others’.

In order to capture some temporal features of the DMM, we propose a simple temporal-pyramid method to extend DMM. We recursively segment the action into several parts along the time axis and then apply the DMM calculation to each part respectively, i.e, if we use the dichotomy to segment the depth sequence, the number of DMMs from top to bottom in the pyramid will be $1, 2, 4, \cdots, 2^{h-1}$ where h is the hierarchy label. Figure 2 shows a DMM-Pyramid architecture which describes the action “pick up & throw” more closely. In this case, we can see that sub-actions in the complex action can be observed clearly with the pyramid growing. So we believe that using DMM-Pyramid as input will improve the classification performance.

On the other hand, the "not-too-deep-hierarchy" pyramid will not increase the computational complexity. In the process of DMM generation, calculating the motions between each two consecutive depth maps and stacking them are most time-consuming. But the work will not be repeated during the pyramid generation because we can directly get high-level DMMs by overlapping low-level DMMs together (e.g. overlapping two images in Figure 2(b) can get the image in Figure 2(a)).

3.3 CNN architecture

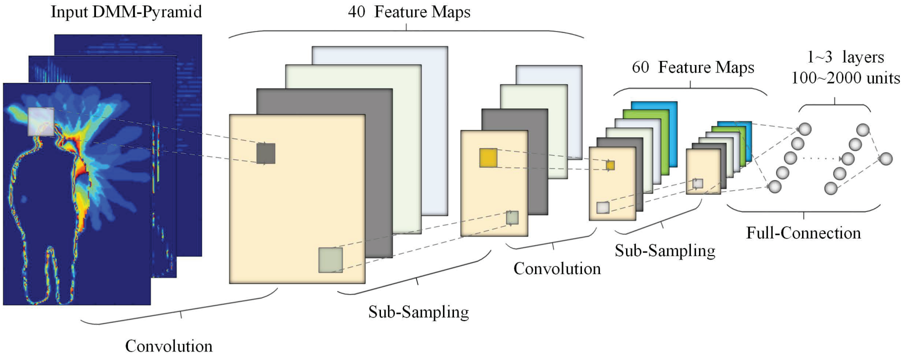


Fig. 3. Our 2D-CNN architecture for human action recognition with DMM-Pyramid as input. The architecture consists of two convolution layers, two sub-sampling layers, and one full connection part consisting of 1-3 layers.

Since all the videos have been organized into DMM-Pyramids, we can simply apply the model similar to the one introduced by LeCun et al [23] to the classification task. The final architecture is illustrated in figure 3.

Including the input layer, the architecture consists of 7-9 layers. We can observe the performance when the full connected multi-layer perceptron part varies. The shape of the filters in two convolution layers is alternative and the poolsize of sub-sampling layers is fixed to 2×2. The size of DMMs is rescaled to 100×100. The first convolution layer consists of 40 feature maps of size 88×88 pixels with the 13×13 filter shape. After max-pooling, the layer connected to the first convolution layer is composed of 40 feature maps of size 44×44 . Following the same principle, the second convolution layer has 60 feature maps of size 34×34 and the filter shape is 11×11. Then the following sub-sampling layer has 60 feature maps of size 17×17. The layers of the multilayer perceptron part is not fixed. Finally, we can instantiate the network by using a logistic regression layer in the end of the whole architecture.

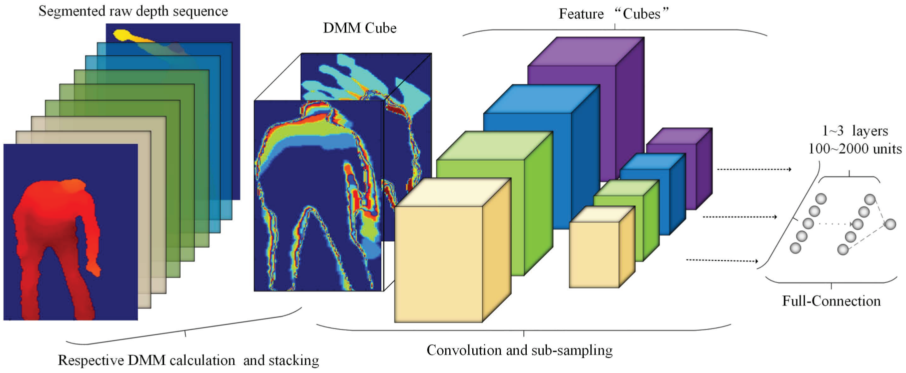

Segmented raw depth sequence
DMM Cube
Feature "Cubes"
1~3 layers
100~2000 units
Full-Connection
Respective DMM calculation and stacking
Convolution and sub-sampling

Fig. 4. Our 3D-CNN architecture for human action recognition with DMM cube as input. The architecture consists of two convolution layers, two sub-sampling layers, and one full connection part consisting of 1-3 layers.

3.4 Strategies for Performance Improvement

Since the benchmarks for human action recognition with depth cameras are usually not large-scale, our architecture's size seems to be much larger than some famous ones. The implementation of the convolution and parameters updating is based on Theano [24] so that we can use fast GPUs to accelerate our experiments. We employ the regularization here to overcome the overffting. L1-norm and L2-sqrt are added in the cost function. More anecdotally, we find that the output probability distribution for each action class changes with the depth of the architecture. So it is plausible that combining the results of single architectures with different depth will improve the performance. Our final strategy is that training three models with different depth (7-9 in this paper) and combining their outputs after each N iterations. The inverse of validation errors of each model are set as weights. In fact, our experiment in section 5 has proved that the combination can both improve the precision and reduce training time.

4 A 3D-CNN Architecture

Compared with 2D CNN, 3D convolution is more straightforward to handle video inputs (depth maps sequence in this paper). The extended convolution in time dimension help us to learn spatio-temporal features automatically. The architecture is shown in figure 4.

4.1 DMM Segmentation and Stacking

Unlike 2D model, the inputs for 3D model need to keep information for the convolution operation on the temporal axis. Single images are no longer applicable here. Our preprocessing method is dividing the depth sequence in a clip evenly into N parts and applying DMM calculation to these parts respectively. So we get N segmented DMMs and they have temporal ordinal relationship between each other. Then we stack these contiguous DMMs together to form a DMM cube (with size $Width \times Height \times N$) and convolute a 3D kernel to it. An example is shown in the left part of figure 4. From another perspective, the final cube is just composed of stacking a bottom layer of a DMM-Pyramid (The DMM-Pyramid-2 in figure 2 can form a usable cube and N is 4 in this case).

4.2 CNN architecture

Regardless of the difference in dimension, the 3D architecture is almost same as the 2D one. The 3D kernel we used is same as the one in [11]. In our experiment, the input layer gets cubes with size $50\times50\times8$, then the 3D filter shape is $7\times7\times3$ so that the 40 "feature cubes" in the first convolution have the size $44 \times 44 \times 6$. We apply 2×2 max-pooling (in the spatial dimension) on each of the feature cubes in the sub-sampling layer and the cubes' size is reduced to $22 \times 22 \times 6$. The next 3D filter shape is $8 \times 8 \times 4$ and sub-sampling size is 3×3. So the size of final input for the full connected multi-layer perceptron part is $60 \times 5 \times 5 \times 3$ (60 filters). The last part is exactly the same as the full connection part in 2D-CNN architecture.

5 Experimental results and discussion

In this section, we show our experimental results produced by applying our method to the public domain MSR Action3D/Gesture3D datasets and compare them with the existing methods.

5.1 Action Recognition on MSR Action3D dataset

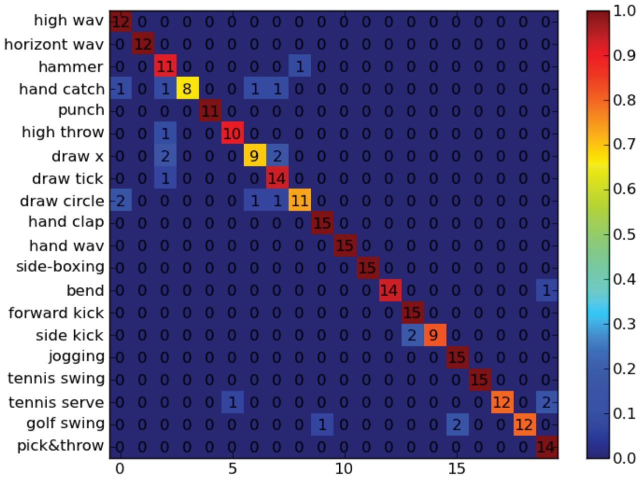
There are 20 action types in the MSR Action3D dataset. Each action type contains 10 subjects and they performed the same action 2-3 times for each subject. 567 depth map sequences are provided in total and 10 of them are abandoned because the skeletons are either missing or too erroneous. The resolution is 320×240. Each depth map sequence we used is regarded as a clip. We compare our method with the state-of-the-art methods on the cross-subject test setting [1], where half subjects are used for training and the rest ones are used for testing.

The performances of our method are shown in Table 1. Intuitively, we see that our work has shown a significant improvement comparing to other methods. The two results of our methods in this table are achieved under the optimal parameters and the combination strategy which is described in section 3.4. Using 2D-CNN with DMM-Pyramid as inputs we obtain the accuracy 91.21%. It shows that the DMM-Pyramid is able to retain enough original information of the video clip and the CNN successfully learns high-level features from it. On the other hand, the 3D-CNN's performance (86.08%) is not as good as expected. We

Table 1. Recognition accuracies (%) comparison based on MSR Action3D dataset

Method	Accuracy %
2D-CNN with DMM-Pyramid	**91.21**
3D-CNN with DMM-Cube	86.08
HON4D + D_{disc}[22]	88.89
Jiang et al. [21]	88.20
Jiang et al. [20]	86.50
Yang et al. [16]	85.52

believe that the resolution 50 × 50 for the entire human motion is too small. It will lose much details of body shape and the temporal features in the dataset may not be as important as spatial features.



high wav	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
horizont wav	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hammer	0	0	11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
hand catch	1	0	1	8	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
punch	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
high throw	0	0	1	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
draw x	0	0	2	0	0	0	9	2	0	0	0	0	0	0	0	0	0	0	0	0
draw tick	0	0	1	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0
draw circle	2	0	0	0	0	0	1	1	11	0	0	0	0	0	0	0	0	0	0	0
hand clap	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
hand wav	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
side-boxing	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0
bend	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	1
forward kick	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
side kick	0	0	0	0	0	0	0	0	0	0	0	0	0	2	9	0	0	0	0	0
jogging	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
tennis swing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0
tennis serve	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	12	0	2
golf swing	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	12	0
pick&throw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14

Fig. 5. The overall confusion matrix for the fixed cross subject test.

Figure 5 shows the confusion matrix of the 2D-CNN (91.21%). The error distribution made by our architecture seems to be more uniform than [21, 22]. Many classification errors occur if two actions are too similar to each other, such as “hand catch” and “high throw” in [21] (25.00%) or “draw X” and “draw circle” in [22] (46.7%). By contrast, the lowest recognition accuracy for single action class in our experiment is 66.67% (“hand catch”). In fact, our method works very well for almost all of the actions.

Figure 6 shows the pace of decline of different models’ error rate. The depth of these models ranges from 7 to 9. The error rate decreases fast in the first 10 to 15 iterations and then becomes very stable. For a single model, the one with

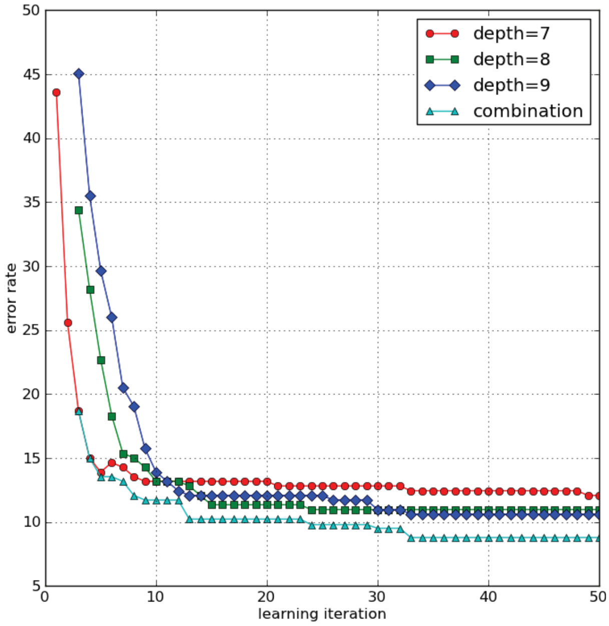


Fig. 6. The pace of decline of different models' error rate.

depth of 9 performs best and takes only 30 iterations to achieve a better result (89.02%) than HON4D [22] and Actionlet Ensemble [21]. On the contrary, the most shallow model reaches 88.28% at iteration 49 and will take another 150 iterations to reach 88.65%. At the same time you may see that the combination of three models touches the 90% line at amazing iteration 13 and the best result showed in table 1 is achieved at iteration 33. It take no more than 30 minutes to learn parameters while the single model cannot reach the accuracy by using days of time. Another interesting thing is that increasing the number of hidden units in the multilayer perceptron part can also slightly speed up the error rate decreasing but a single iteration time will be longer because there are more parameters to train.

5.2 Gesture Recognition on MSR Gesture3D dataset

There are 12 dynamic American Sign Language (ASL) gestures in MSR Gesture3D dataset: “bathroom”, “blue”, “finish”, “green”, “hungry”, “milk”, “past”, “pig”, “store”, “where”, “j”, “z”. All of the gestures were captured by a Kinect device. There are 336 files in total, each corresponding to a depth sequence just like MSR Action3D dataset. We follow the experiment setup in [17]. Table 2 shows that our architecture outperforms the state of the art methods. We also notice that the 3D-CNN architecture performs much better here than in Action3D dataset. We believe that low resolution will not cause too much impact on the recognition of local motion like gestures.

Table 2. Recognition accuracies (%) comparison based on MSR Gesture3D dataset

Method	Accuracy %
2D-CNN with DMM-Pyramid	**94.35**
3D-CNN with DMM-Cube	92.25
HON4D + D_{disc}[22]	92.45
Jiang et al. [21]	88.50
Yang et al. [16]	89.20

6 Conclusion

In this paper, we presented two deep architectures based on convolutional neural networks for human action recognition from depth sequences. In 2D-CNN based architecture, we proposed a DMM-Pyramid method to stack contiguous frames together into still images and convert action recognition to "image classification" work. In 3D-CNN based architecture, we use the DMM Cube as the inputs for the networks and expect to learn more temporal features. Both of the architectures aim to learn spatial and temporal features automatically from the raw inputs without complex preprocessing. We also find that applying a linear weighted combination to CNN models with different depth can significantly improve the precision and reduce learning time. Our experiments on some widely-used and challenging datasets show that the proposed architectures give competitive results, among the best of related work, both on MSR Action3D (91.21%) and MSR Gesture3D (94.35%). Our methods are easy to implement based on the open-source python library Theano and anyone can reproduce the experiment to achieve the good (or even better) results following the setup described in section 5.

Furthermore, although the MSR Action3D dataset remains to be the most widely used dataset for human action recognition with depth inputs, there are some other challenging datasets (especially more realistic) for us to verify our architectures' genericity, e.g MSR DailyActivity3D dataset [21] and Subtle Walking From CMU Mocap Dataset [25]. We also plan to do a large-scale experiment to confirm our CNN models' performance in practice.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61321491 and 61272218.

References

1. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE (2010) 9–14
2. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56** (2013) 116–124

3. Li, W., Zhang, Z., Liu, Z.: Expandable data-driven graphical modeling of human actions based on salient postures. Circuits and Systems for Video Technology, IEEE Transactions on **18** (2008) 1499–1510
4. Zhang, S.: Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. Optics and lasers in engineering **48** (2010) 149–158
5. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: A convolutional neural-network approach. Neural Networks, IEEE Transactions on **8** (1997) 98–113
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
7. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 1915–1929
8. Kavukcuoglu, K., Sermanet, P., Boureau, Y.L., Gregor, K., Mathieu, M., Cun, Y.L.: Learning convolutional feature hierarchies for visual recognition. In: Advances in neural information processing systems. (2010) 1090–1098
9. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 3642–3649
10. Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., Barbano, P.E.: Toward automatic phenotyping of developing embryos from videos. Image Processing, IEEE Transactions on **14** (2005) 1360–1371
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 221–231
12. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: Human Behavior Understanding. Springer (2011) 29–39
13. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014)
14. Ahad, M.A.R., Tan, J.K., Kim, H., Ishikawa, S.: Motion history image: its variants and applications. Machine Vision and Applications **23** (2012) 255–281
15. Han, J., Bhanu, B.: Individual recognition using gait energy image. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28** (2006) 316–322
16. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on Multimedia, ACM (2012) 1057–1060
17. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, IEEE (2012) 1975–1979
18. Wu, D., Zhu, F., Shao, L.: One shot learning gesture recognition from rgbd images. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE (2012) 7–12
19. Yang, X., Tian, Y.: Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE (2012) 14–19
20. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: Computer Vision–ECCV 2012. Springer (2012) 872–885

21. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1290–1297
22. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 716–723
23. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361** (1995)
24. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a cpu and gpu math expression compiler. In: Proceedings of the Python for scientific computing conference (SciPy). Volume 4. (2010) 3
25. Han, L., Wu, X., Liang, W., Hou, G., Jia, Y.: Discriminative human action recognition in the learned hierarchical manifold space. Image and Vision Computing **28** (2010) 836–849