HIERARCHICAL DROPPED CONVOLUTIONAL NEURAL NETWORK FOR SPEED INSENSITIVE HUMAN ACTION RECOGNITION

Fanyang Meng[1,2]*, Hong Liu*[1,*]*, Yongsheng Liang*[2]*, Mengyuan Liu*[3]*, Wei Liu*[2]

[1]Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School, China
[2]Shenzhen Institute of Information Technology, China
[3]Nanyang Technological University, Singapore
fymeng@pkusz.edu.cn hongliu@pku.edu.cn {liangys,liuwei}@sziit.edu.cn liumengyuan@ntu.edu.sg

ABSTRACT

Human action recognition using skeleton data has lots of potential applications in content-based action retrieval and intelligent surveillance, with wide usage of depth sensors and robust skeleton estimation algorithms. Previous methods describe spatial temporal skeleton joints as a compact color image and then use Convolutional Neural Network (CNN) to extract more discriminative deep features. However, these methods ignore the effect of speed variation, which is a common phenomenon and can bring severe intra-varieties to same types of actions. To solve this problem, this paper presents a novel hierarchical dropped CNN architecture, which is constructed in two stages. Dropped CNN (d-CNN) is firstly developed to extract deep features from a probabilistic speed insensitive color image. This image expresses both spatial distributions and temporal evolutions of skeleton joints meanwhile avoids the effect of speed variations. To enhance the temporal discriminative power, we extend d-CNN to a hierarchical structure (h-CNN), where multiple scales of temporal information are encoded. Extensive experiments on benchmark MSRC-12 dataset and the largest NTU RGB+D dataset verify the effectiveness and robustness of the proposed method.

Index Terms— Human action recognition, Skeleton, CNN, Dropout

1. INTRODUCTION

Human action recognition has been widely explored, bringing applications to many fields, such as content-based action retrieval [1], intelligent surveillance [2], gaming [3] and so on. The first attempt of this task uses RGB data, since RGB sensor is cheap and has been used in various scenarios. Since RGB sensor cannot capture depth information, it is rather difficult for algorithms to detect human bodies from cluttered background. Moreover, the lost of depth data brings ambiguities for distinguishing similar actions. With the progress of depth sensor, i.e., Microsoft Kinect, researchers begin using depth data for human action recognition. Compared with RGB data, human bodies can be segmented from backgrounds more easily, since the complex and confusing textures or illuminations are ignored by depth sensor. More importantly, additional information from depth data provides a new view to distinguish actions whose appearances are similar from the view of X-Y plane but different in the depth (Z axis) direction. The drawbacks of depth data are mainly two folds. First, the depth data contains jumping noises. Second, depth data is usually redundant for mapping a complex depth sequence to a simple action label. Recently, robust skeleton estimation algorithms can extract skeleton joints from depth data in realtime, which opens a new way for understanding human actions using 3D skeleton data. Compared with depth data, skeleton joints estimated by any robust algorithm [4] is more compact and suffers less from jumping noise.

It is still a challenging problem to describe spatial temporal skeleton joints. Inspired by the impressive achievements of Convolutional Neural Network (CNN) in the field of image classification, recent work [5] describes spatial temporal skeleton joints as a compact color image and use CNN to extract more discriminative deep features. As CNN is originally designed for encoding spatial information, the key issue is to express skeleton sequences as images. Specifically, 3D coordinates of skeleton joints are divided into three channels (X, Y, Z). For each channel, the coordinates of each frame are arranged as a column vector. Then, all vectors are concatenated as a matrix according to the temporal order. Finally, three matrices respectively denote three channels of a color image. Based on the obtained color image, pre-trained CNN models, such as AlexNet, ResNet, VGGNet, can be used

This work is supported by Natural Science Foundation of China (NSFC, No.U1613209,61340046,61673030), Natural Science Foundation of Guangdong Province (No.2015A030311034), Scientific Research Project of Guangdong Province (No.2015B010919004), Specialized Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (No.ZLZBCXLJZI20160729020003), Scientific Research Project of Shenzhen City (No.JCYJ20170306164738129), Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (No.ZDSYS201703031405467), Scientific Research Platform Cultivation project(PT201704). Hong Liu is the corresponding author.

978-1-5386-1737-3/18/$31.00

to extract deep features, which implicitly encode both spatial and temporal information of skeleton joints.

Above pipeline provides a simple yet efficient way to represent skeleton sequences. However, it ignores the effect of speed variation, which is a common phenomenon and can bring severe intra-varieties to same types of actions. Different habits of humans induce the speed variations. Even the same person may use different speeds when repeating same type of action. Regardless of speed variations, two person who are waving hands with different frequencies should naturally be treated as perform the same action "waving".

In this paper, we propose a hierarchical dropped CNN method, which eliminates the effect of speed variations. Fig. 1 shows the proposed pipeline. After encoding a skeleton sequence as a color image, we further use dropout layer to obtain a probabilistic speed insensitive image. Note that dropout layer is usually used after full connected layer. Here, we try the dropout layer for feature extraction. In the training stage, the dropout layer generates a set of images, each of which is according to a specific speed. In this way, the trained CNN model can adapt to actions with different speed variations. Here, we call the image generated by dropout layer as probabilistic speed insensitive image for the reason that the image is unbiased to any specific speed in the training stage. Above trained CNN model is called as Dropped CNN (d-CNN), which is robust to speed variations but has limited temporal discriminative power since only fixed scale of temporal information is considered. To this end, d-CNN extended to a hierarchical structure (h-CNN), where multiple scales of temporal information are encoded. We pre-define a set of temporal scales and then train corresponding d-CNN models in an end-to-end manner. The decision-level fused deep features encode the multiple scale information.

We summarize main contributions of this paper as threefold. First, we propose an end-to-end hierarchical dropped CNN model to extract distinctive and speed insensitive deep features for skeleton-based human action recognition. Detailed structures of d-CNN and hd-CNN are implemented to efficiently encode both spatial and temporal relations of skeleton joints. Second, we analyze the potential usage of d-CNN and hd-CNN for tackling data with variable length. Third, we evaluate our method on the skeleton-based action recognition task and achieve state-of-the-art recognition accuracies on MSRC-12 and the largest NTU RGB+D dataset.

2. RELATED WORK

Estimating skeleton joints from depth images discussed in [4] provides a more intuitive way to perceive human actions. Skeleton based approaches utilize the high-level skeleton information extracted from depth video sequences. In [6], skeleton joint locations were placed into 3D spatial bins to build histograms of 3D joint locations (HOJ3D) as features for action recognition. Yang et al. [7] adopted the differences of joints in temporal and spatial domains to encode the dynamics of joints and then obtain the EigenJoints by applying Principal Component Analysis (PCA) to joint differences. The EigenJoints contain less redundancy and noises, compared with original joints. Zanfir et al. [8] provided a non-parametric Moving Pose (MP) framework, which considers more features like position, speed and acceleration of joints. In [9], an evolutionary algorithm was used to select the optimal subset of skeleton joints based on the topological structure of a skeleton leading to improved recognition rates. In [10], human actions were modeled by a spatio-temporal hierarchy of bio-inspired 3D skeletal features. Linear dynamical systems were employed to learn the dynamics of these features. Kerola et al. [11] constructed a spatial temporal graph by linking joints in consecutive skeletons, where edge weights are calculated by distances. A spectral graph wavelet transform (SGWT) was applied on the 3D skeleton graph to create an overcomplete representation. In [12], Cai et al. developed a novel action attribute mining method, where an attribute space was built by the geometry transformation between body parts. In [13], a body part-based skeleton representation was proposed to model the relative geometry between body parts. Then, human actions were modeled as curves using a Lie group $SE(3) \times ... \times SE(3)$, which explicitly models the 3D geometric relationships among human body parts. In [14], the skeleton was divided into five parts, which were used as inputs for five bidirectional recurrent neural networks (BRNNs). Then, the representations from the subnets were fused in a hierarchical way to be the inputs to higher layers. Since recurrent neural network (RNN) can model the long-term contextual information of temporal sequences, the proposed end-to-end hierarchical RNN achieved high performances on the task of skeleton-based action recognition. Although some of the skeleton-based approaches obtain high recognition performance, skeleton-based methods are not applicable for applications where skeleton information is not available.

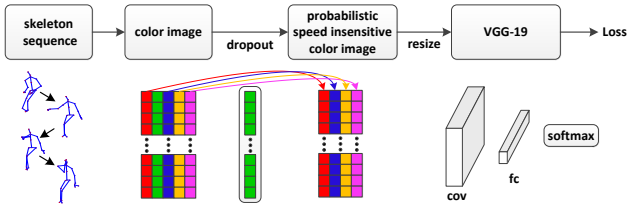

Fig. 1. Dropped CNN structure

3. HIERARCHICAL DROPPED CONVOLUTIONAL NEURAL NETWORK FOR ACTION RECOGNITION

3.1. Color Image

Let $\{\{(x_n^t, y_n^t, z_n^t)\}_{n=1}^N\}_{t=1}^T$ be a skeleton sequence with T frames. Each frame contains N skeleton joints. N is determined by skeleton estimation algorithm and varies from d-

ifferent datasets. For example, in the currently largest NTU RGB+D dataset, N is equal to 25. The coordinates of skeleton joints depend on a global coordinate system. The translation among different skeleton sequences is not directly related to actions. Therefore, we remove the translation by moving the origin of global coordinate system to the central of each skeleton. When recording skeleton sequences, the distances between human bodies and the depth sensor are not strictly the same. In other words, different skeleton sequences have specific scales, which bring intra-varieties to same types of actions. To this end, we normalize the coordinates. Specifically, $\{\{x_n^t\}\}_{n=1}^N\}_{t=1}^T$, $\{\{y_n^t\}\}_{n=1}^N\}_{t=1}^T$ and $\{\{z_n^t\}\}_{n=1}^N\}_{t=1}^T$ are restricted to change from 0 to 1.

Let [R,G,B] be the color image which represents the preprocessed skeleton sequence. Here, R, G, B denote three matrices, which contain N rows and T columns. For the n-th row and the t-th column, values on R, G and B are equal to x_n^t, y_n^t and z_n^t, respectively. To facilitating the usage of pre-trained CNN models, such VGG-19, we further resize the color image to the size of $[224, 224]$. Fig. 2 (a) is a skeleton sequence. We use non-linear sampling to generate two sequences from original sequence. These two sequences indicate same type of action performed with different speeds. Fig. 2 (b) and (c) are the corresponding color image features. Fig. 2 (d) is the absolute different between two color images. We draw two observations from Fig. 2. First, color image is a compact feature shows spatial temporal discriminative power to represent action. Second, color image is sensitive to the speed variations.

3.2. Dropped CNN

Dropout layer Deep networks usually need large amount of data for training. However, in most cases, training data is limited, which induces the overfitting of deep network. To solve this problem, dropout layer is proposed to randomly ignore some neurons in each training epoch. Though this scheme is simple, it forces different neurons to learn different aspects of training data. In this way, the overfitting phenomenon is suppressed. One parameter r is used to control the ignorance rate. When r is too large, the deep network will trend to overfitting. When r is too small, the deep network will be hard to converge. Therefore, a proper value of r to ensure the performance of the dropout layer.

Dropped CNN Traditionally, the dropout layer is used among different layers of deep networks. The effect of dropout layer on feature processing is usually ignored. To eliminate the effect of speed variations on color image feature, we use dropout layer for feature conversion. The corresponding proposed dropped CNN (d-CNN) is shown in Fig. 1, where the dropout layer is used to convert the color image to a probabilistic speed insensitive color image. To simplify the case, a skeleton sequence in Fig. 1 is supposed to contain five frames, which are colored in red, green, blue, yellow and pink. Let

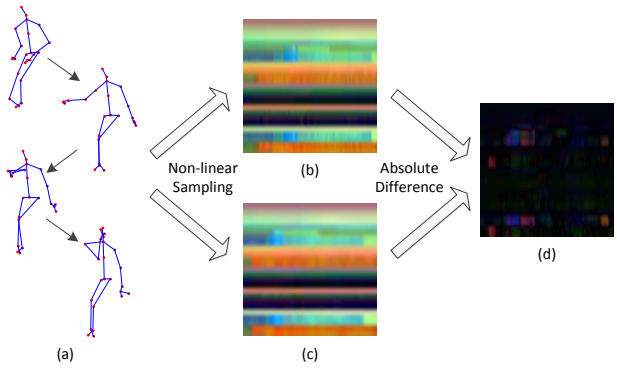

Fig. 2. Sensitivity of color image feature to speed variations

the parameter r equal to 0.2. The dropout layer randomly ignores 20 percent (0.2) of all frames. In this case, the frame colored in green is ignored and other four frames are concatenated according to original order. Since the column number of the generated color image is changed, we further resize it to $[224, 224]$ for the input of VGG-19 network.

Relation to data augmentation Despite using dropout layer between different layers, data augmentation method is also a popular way to alleviate the overfitting. Traditional data augmentation includes random cropping, horizontal flipping, adding gaussian noise and so on. The generated color images using dropout layer in our method can be treated as a new type of data augmentation method. Random cropping is not suitable for our task, since the horizontal information of color image has strict structure, namely the order of skeleton joints. Another difference is that random cropping will crop a patch of continuous region from color image. In other words, the cropped patch is according to the same specific speed of original action. While, our method can generate new color image features according to various speed variations. Generally, our method is specifically designed for augmenting actions with various speeds, which has not been researched yet.

Tackling data with variable length Previous works usually use RNN/LSTM to tacking data with variable length. Recently, some works directly resize data to fixed length and then use CNN to extract deep features. We argue that CNN may perform better than RNN/LSTM if we properly arrange data with variable length as suitable inputs for CNN. Our d-CNN model has two merits to tacking data with variable length. First, different lengthes of inputs are sampled to fixed sizes, therefore facilitates the usage of CNN. Second, in different training epoch, different data are sampled, which avoid the loss of input data.

3.3. Hierarchical CNN

Let parameter r_m be the ignorance rate of the dropout layer. For skeleton sequence $\{\{(x_n^t, y_n^t, z_n^t)\}_{n=1}^N\}_{t=1}^T$, the generated probabilistic speed insensitive color image will has N rows and $\lceil T * (1 - r_m) \rceil$ columns. The number of columns reflects

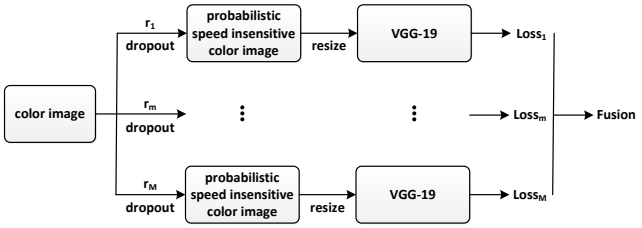

Fig. 3. Hierarchical CNN structure

the scale of temporal domain. Single scale will limit the discriminative power of extracted deep feature. To this end, we set the ignorance rate to $\{r_m\}_{m=1}^{M}$, and train an end to end hierarchical CNN model, which is shown in Fig. 3. Note that we use pre-trained parameters for VGG-19 network except for the last full connection layer.

For an input sequence $\mathcal{I}^k$, we obtain a series of color images: $\{\mathbf{I}_m^k\}_{m=1}^{M}$. Mean removal is adopted for all input images to improve the convergence speed. Then, each color image is processed by a CNN. For the image $\mathbf{I}_m^k$, the output Υ_m of the last fully-connected (fc) layer is normalized by the softmax function to obtain the posterior probability:

$$prob(l \mid \mathbf{I}_m^k) = \frac{e^{\Upsilon_m^l}}{\sum_{j=1}^{L} e^{\Upsilon_m^j}}, \tag{1}$$

which indicates the probability of image $\mathbf{I}_m^k$ belonging to the l-th action class. L is the number of total action classes.

The objective function of our model is to minimize the maximum-likelihood loss function:

$$\mathcal{L}(\mathbf{I}_m) = -\sum_{k=1}^{K} ln \sum_{l=1}^{L} \delta(l - s_k)\, prob(l \mid \mathbf{I}_m^k), \tag{2}$$

where function δ equals one if $l = s_k$ and equals zero otherwise, s_k is the real label of $\mathbf{I}_m^k$, K is the batch size. For sequence $\mathcal{I}$, its class score is formulated as:

$$score(l \mid \mathcal{I}) = \frac{1}{M} \sum_{m=1}^{M} prob(l \mid \mathbf{I}_m), \tag{3}$$

where $score(l \mid \mathcal{I})$ is the average of the outputs from all ten CNN and $prob(l|\mathbf{I}_m)$ is the probability of image $\mathbf{I}_m$ belonging to the l-th action class.

4. EXPERIMENTS

4.1. Datasets and Settings

NTU RGB+D dataset [15] contains 60 actions performed by 40 subjects from various views (Fig. 4 (a)), generating 56880 skeleton sequences. This dataset also contains noisy skeleton joints (see Fig. 4 (b)), which bring extra challenge for recognition. Following the cross subject protocol in [15], we split the 40 subjects into training and testing groups. Each group contains samples captured from different views performed by 20 subjects. For this evaluation, the training and testing sets have 40320 and 16560 samples, respectively. Following the cross view protocol in [15], we use all the samples of camera 1 for testing and samples of cameras 2 and 3 for training. The training and testing sets have 37920 and 18960 samples, respectively.

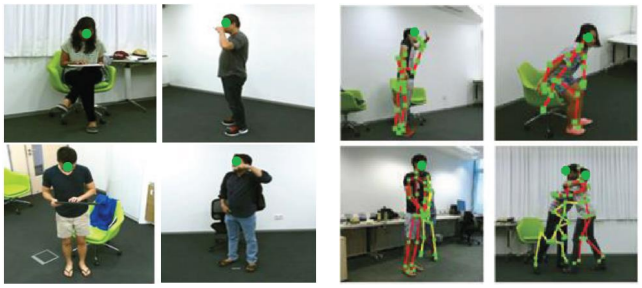

Fig. 4. Snaps from NTU RGB+D dataset

Table 1. Evaluation on MSRC-12 dataset

Method	Accuracy
CNN	91.07%
CNN+Random Cropping	87.84%
d-CNN (r=0.1) (***ours***)	89.78%
d-CNN (r=0.2) (***ours***)	91.02%
d-CNN (r=0.3) (***ours***)	89.28%
d-CNN (r=0.4) (***ours***)	90.99%
h-CNN	93.79%
hd-CNN (***ours***)	94.59%

MSRC-12 dataset [16] contains 594 sequences, i.e. 719359 frames (approx. 6 hour 40 minutes), collected from 30 people performing 12 gestures. This is a single view dataset, i.e., action samples are captured from a single view. Therefore, the sequence-based transform method is not used to implement our method on this dataset. Following the cross-subject protocol in [17], we use sequences performed by odd subjects for training and even subjects for testing.

Implementing details In our model, each CNN contains five convolutional layers and three fc layers. The first and second fc layers contain 4096 neurons, and the number of neurons in the third one is equal to the total number of action classes. Filter sizes are set to 11×11, 5×5, 3×3, 3×3, 3×3. Local Response Normalisation (LRN), max pooling and ReLU neuron are adopted and the dropout regularisation ratio is set to 0.5. The network weights are learned using the mini-batch stochastic gradient descent with the momentum value set to 0.9 and weight decay set to 0.00004. Initial learning rate is set to 0.001 and then divided by 10 every 20 epoches. The maximum training cycle is set to 80. In each cycle, a mini-batch of 128 samples is constructed by randomly sampling 128 images from training set. The implementation is based on pyTorch with one TITAN X card and 16G RAM.

Table 2. Evaluation on NTU RGB+D dataset

Method	Cross Subject	Cross View
CNN	81.77%	90.00%
CNN+Random Cropping	79.41%	87.65%
d-CNN (r=0.1) (***ours***)	82.23%	90.40%
d-CNN (r=0.2) (***ours***)	81.38%	90.13%
d-CNN (r=0.3) (***ours***)	81.40%	90.05%
d-CNN (r=0.4) (***ours***)	81.44%	89.99%
h-CNN	83.21%	91.15%
hd-CNN (***ours***)	84.33%	92.21%

Table 3. Comparisons on MSRC-12 dataset

Method	Accuracy
ELC-KSVD [18]	90.22%
Cov3DJ [19]	91.70%
ConvNets [5]	84.46%
JTM [17]	93.12%
hd-CNN (***ours***)	94.59%

4.2. Effectiveness of the Proposed Method

To simplify analysis, we use following abbreviations. *CNN* denotes using color image feature to represent a skeleton sequence and using a pre-trained VGG-19 network for extracting deep features. *CNN+Random Cropping* denotes applying random cropping method to do data augmentation. *d-CNN (r=0.1)* denotes using our d-CNN network with parameter $r = 0.1$. *h-CNN* denotes using *h-CNN* network which fuses four different *CNN* networks, *hd-CNN* denotes using our *hd-CNN* network which fuses *d-CNN* networks with $r = 0.1, 0.2, 0.3, 0.4$.

Parameter evaluation As shown in Table 1 and 2, the gap between the *d-CNN* and the *CNN* is not obvious, for example, only in the case of $r = 0.1$, the performance of the *d-CNN* is slightly better than the *CNN*. It probably can be explained that with the increases of *r*, the more temporal information is discarded. It infers that the *d-CNN* can barely learn detail features over each time slot. Even so, the performance of *d-CNN* is still very close to the *CNN*.

d-CNN versus hd-CNN The performance of the *hd-CNN* is better than the *d-CNN*, for example, the accuracy of the *h-CNN* increases to 2.21% on NTU RGB+D dataset with cross view. It means the accuracy of our method can be improved when multiple scales of temporal information are considered.

h-CNN versus hd-CNN The performance of the *hd-CNN* is better than the *h-CNN*, It verifies that considering the speed invariance is necessary for human action recognition.

Data augmentation The *CNN* is better than the *CNN+Random Cropping* in terms of accuracy. This means that the data augmentation using random cropping is not suitable for skeleton sequence color image. It can be explained that when the skeleton sequence color image is cropped with continuous region, the temporal of human action will be lost.

Table 4. Comparisons on NTU RGB+D dataset

Method	Cross Subject	Cross View
SNV [20]	31.82%	13.61%
HOG^2 [21]	32.24%	22.27%
Dynamic Skeletons [22]	60.23%	65.22%
HBRNN-L [14]	59.07%	63.97%
Deep RNN [15]	56.29%	64.09%
Deep LSTM [15]	60.69%	67.29%
Part-aware LSTM [15]	62.93%	70.27%
ST-LSTM [23]	61.70%	75.50%
ST-LSTM+TG [23]	69.20%	77.70%
View-invariant CNN [24]	80.03%	87.21%
Two-Stream RNN/CNN [25]	83.74%	93.65%
hd-CNN (***ours***)	84.33%	92.21%

4.3. Comparisons to the State-of-the-Art Methods

Table 3 and 4 shows the performances of various methods on MRSC-12 and NTU RGB+D dataset. As shown in Table 3 and 4, The performance of the *h-CNN* are better than other algorithms. It is worth noting that the View-invariant CNN [20] in Table 4 is most related to our visualization method. Although view variations on spatio-temporal locations of skeleton joints is effectively eliminated in the View-invariant CNN, the *h-CNN* is still better than the View-invariant CNN. It probably can be explained that compared with the View-invariant CNN [24], the *h-CNN* not only can leaning spatial-invariant features automatically, but also can learning temporal-invariant features effectively.

5. CONCLUSION

In this paper, we demonstrate the effect of speed variation on skeleton sequence color image. To address the problem, we propose a novel hierarchical dropped CNN architecture. Dropped CNN (d-CNN) is firstly developed to extract deep features from a probabilistic speed insensitive color image. This image expresses both spatial distributions and temporal evolutions of skeleton joints meanwhile avoids the effect of speed variations. Then, we extend d-CNN to a hierarchical structure (h-CNN) to encoding multiple scales of temporal information. Extensive experiments on benchmark MSRC-12 dataset and the currently largest NTU RGB+D dataset verify the effectiveness and robustness of our proposed method. In the future, we will encode multi-scale information via one CNN network.

6. REFERENCES

[1] Min Chun Hu, Chi Wen Chen, Wen Huang Cheng, Che Han Chang, Jui Hsin Lai, and Ja Ling Wu, "Real-time human movement retrieval and assessment with Kinect sensor.," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 742–753, 2014.

[2] Georgios Mastorakis and Dimitrios Makris, "Fall detection system using Kinect's infrared sensor," *Journal of Real-Time Image Processing*, vol. 9, no. 4, pp. 635–646, 2014.

[3] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," in *CVPRW*, 2012, pp. 7–12.

[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011, pp. 1297–1304.

[5] Yong Du, Yun Fu, and Liang Wang, "Skeleton based action recognition with convolutional neural network," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. nov 2015, pp. 579–583, IEEE.

[6] Lu Xia, Chia Chih Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *CVPRW*, 2012, pp. 20–27.

[7] Xiaodong Yang and Ying Li Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *CVPRW*, 2012, pp. 14–19.

[8] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *ICCV*, 2013, pp. 2752–2759.

[9] Alexandros Andre Chaaraoui, Jos Ramn Padilla-Lpez, Pau Climent-Prez, and Francisco Flrez-Revuelta, "Evolutionary joint selection to improve human action recognition with rgb-d devices," *Expert Systems with Applications*, vol. 41, no. 3, pp. 786–794, 2014.

[10] Rizwan Chaudhry, Ferda Ofli, Gregorij Kurillo, Ruzena Bajcsy, and Rene Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *CVPRW*, 2013, pp. 471–478.

[11] Tommi Kerola, Nakamasa Inoue, and Koichi Shinoda, *Spectral Graph Skeletons for 3D Action Recognition*, Springer International Publishing, 2014.

[12] Xingyang Cai, Wengang Zhou, and Houqiang Li, "Attribute mining for scalable 3d human action recognition," in *ACM MM*, 2015, pp. 1075–1078.

[13] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *CVPR*, 2014, pp. 588–595.

[14] Yong Du, Wei Wang, and Liang Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015.

[15] Amir Shahroudy, Jun Liu, Tian Tsong Ng, and Gang Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *CVPR*, 2016, pp. 1010–1019.

[16] Simon Fothergill, Helena Mentis, Pushmeet Kohli, and Sebastian Nowozin, "Instructing people for training gestural interactive systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1737–1746.

[17] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *ACM MM*, 2016, pp. 102–106.

[18] Lijuan Zhou, Wanqing Li, Yuyao Zhang, Philip Ogunbona, Duc Thanh Nguyen, and Hanling Zhang, "Discriminative key pose extraction using extended LC-KSVD for action recognition," in *DICTA*, 2014, pp. 1–8.

[19] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *IJCAI*, 2013, pp. 2466–2472.

[20] Xiaodong Yang and Ying Li Tian, "Super Normal Vector for human activity recognition with depth cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, p. 10.1109/TPAMI.2016.2565479, 2016.

[21] Eshed Ohn-Bar and Mohan Trivedi, "Joint angles similarities and HOG^2 for action recognition," in *CVPRW*, 2013, pp. 465–470.

[22] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *CVPR*, 2015, pp. 5344–5352.

[23] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," 2016.

[24] Mengyuan Liu, Hong Liu, and Chen Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[25] Rui Zhao, Haider Ali, and Patrick van der Smagt, "Two-stream RNN/CNN for action recognition in 3D videos," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, sep 2017, pp. 4260–4267.