International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Multi-Task Learning Organize (MTLN) of Skeleton Sequences Based 3D Action Recognition

T. Seshagiri[1] , S. Varadarajan[2]

[1]Scholar, Rayalaseema University, Kurnool, Associate Professor, Shree Institute of Technical Education, Tirupati, Andhra Pradesh, India

[2]Professor, Department of Electronics &Communication Engineering, Svu Engineering College, Tirupati, Andhra Pradesh, India

ABSTRACT

Skeleton sequences provide 3D trajectories of human skeleton joints. The spatial temporal in sequence is very significant for action detection. Considering that deep convolution neural network (CNN) is very influential for feature learning in images, in this paper, we intend to transform a skeleton sequence into an image-based demonstration for spatial temporal information learning with CNN. Specifically, for each channel of the 3D coordinates, we distinguish the sequence into a clip with several gray images, which represent multiple spatial structural information of the joints. Those images are fed to a deep CNN to learn high-level features. The CNN features of all the three clips at the same time-step are concatenated in a feature vector. Each feature vector represents the temporal information of the entire skeleton sequence and one particular spatial relationship of the joints. Then we propose a Multi-Task Learning Network (MTLN) to jointly process the feature vectors of all time-steps in related for action detection. Investigational results clearly show the effectiveness of the proposed new representation and feature learning method for 3D action detection.

Keywords: Temporal pooling of CNN, MTLN, LSTM, HMM and CRF.

I. INTRODUCTION

Human illustration based on 3D skeleton data encodes the complete person body with joints. It is strong to illumination changes and invariant to camera views [9], with the incidence of highly-accurate and reasonable devices, action detection based on 3D skeleton sequence has been attracting growing interest [32, 28, 4, 24, 36, 16, 14]. In this paper, we focus on skeleton-based action detection. Given a skeleton sequence, the temporal dynamics of several frames and the spatial structural information of the skeleton joints in a single frame give important cues for action detection [36]. The most existing works implicitly model the temporal dynamics of skeleton sequences using Hidden Markov Models (HMMs) [31], Conditional Random Fields (CRFs) [26] or Temporal Pyramids (TPs) [28]. To make use of the spatial structure, different features have been investigated, such as histogram of joint positions [32], pair wise relative position [29] and 3D rotation and transformation [28]. In recent times, recurrent neural networks (RNNs) with Long-Short Term Memory (LSTM) neurons [7, 8] have also been used to form the spatial formation [4, 24, and 36] or jointly the spatial and temporal information of skeleton sequences [16]. All of the mentioned works directly operate on the local 3D coordinates of the joints to take out features and learning models. While, the coordinates of the joints are not always accurate, this usually results in poor features. In addition, it is also hard to handle the large temporal variations and complex spatial structures using the native coordinates of the noisy skeleton joints. Considering that CNNs are capable of learning healthy features, in this paper, instead of directly operating on the local 3D coordinates of the joints to extract features, we convert each skeleton sequence to three video clips, and then utilize deep networks to learn features from the clips for action detection.

Mainly, given a skeleton sequence, several joints are selected as the reference joints, which we use to generate multiple set of vectors by separately comparing the reference joints with the others. Three clips corresponding to the 3D coordinate of the vectors are then obtained. Each clip is enclosed multiple frames generated from the different sets of vectors. Every frame of the clips explains the temporal in sequence of the complete skeleton sequence, and includes one particular spatial association among the joints. The whole clips collective multiple frames with the different spatial relationships, providing important information of the spatial formation of the skeleton joints.

Unlike the novel skeleton sequence, which just contains the coordinates of the discrete joints, the generated clips are contains of images. The advantage of the generated clips above the unique skeleton series is that deep CNN models pre-trained with large-scale Image Net [22] can be leveraged to extract representations which are invariant and are insensitive to noise. CNNs are known to learn image features that are vigorous to noise due to the convolution and pooling operators. The learned features are generic and can be transferred to novel tasks from the original tasks [34, 17].

More specifically, each frame of the generated clips is fed to a pre-trained CNN model followed by a temporal pooling layer to extract a CNN feature. Then the three CNN features of the three clips at the related time-step (See Figure 1) are link together in a feature vector. Accordingly, multiple feature vectors are extracted from all the time-steps. Every feature vector represents one exacting spatial association between the joints. All the feature vectors of different time-steps represent the different spatial relationships and there exist intrinsic relationships among them. Therefore, this paper proposes to utilize the intrinsic relationships among different feature vectors for action recognition using a Multi-Task Learning Network. Multi-task learning aims at improving the generalization performance by jointly training multiple related tasks and utilizing their essential relationships [1]. In the proposed MTLN, the categorization of each feature vector is treated as a split task, and the MTLN together learns every one of of the feature vectors and outputs different predictions, every equivalent to one task. All the feature vectors of the same skeleton sequence have the same label as the skeleton sequence. During training, the loss value of each task is individually computed using its own class scores. Then the loss values of all tasks are summed up to define the final loss of the network which is then used to update the network parameters. During testing, the class scores of all tasks are averaged to form the final prediction of the action class. Multi-task learning concurrently solves multiple tasks with weight sharing, which can advance the presentation of individual tasks [1].

II. RELATED WORKS

In this part, we cover the significant journalism of skeleton-based action finding by hand-crafted features and deep learning methods.

Hand-crafted Features In [12], the covariance matrices of the trajectories of the combined positions is computed over hierarchical of time levels to model the skeleton sequences. In [29], the pair wise relative position of each joint with other joints is computed to differentiate every frame of the skeleton sequences, and Fourier Temporal Pyramid is used to replica the temporal patterns. In [33], the pair wise relative positions of the joints are also used to characterize pose features, motion features, and recompense features of the skeleton sequences. Principal Component Analysis is then applied to the normalized features to compute Eigen Joints as representations. In [32], histograms of 3D joint locations are computed to set apart each frame of the skeleton sequences, and HMMs are used to model the temporal dynamics. In [28], the rotations and translations among various body parts are used as representations, and a skeleton sequence is modeled as a curve in the Lie group. The temporal dynamics are modeled with FTP.

Deep Learning Methods In [4], the skeleton joints are separated into five sets equivalent to five body parts. They are fed into five BLSTMs for feature fusion and classification. In [36], the skeleton joints are fed to a deep LSTM at each time slot to learn the inherent co-occurrence features of skeleton joints. In [24], the long-term context representations of the body parts are learned with a part aware LSTM. In [16], both the spatial and temporal information of skeleton sequences are well read with a spatial temporal LSTM. A Trust Gate is also proposed to get rid of noisy joints. This

method achieves the up to date performance on the NTU RGB+D dataset [24].

III. PROPOSED SYSTEM

An overall architecture of the proposed method is shown in Figure 1. The proposed method starts by generating clips of skeleton sequences. A skeleton sequence with an arbitrary number of frames is transformed into three clips corresponding to the different channels of the cylindrical coordinates, as shown in Figure 1(b).

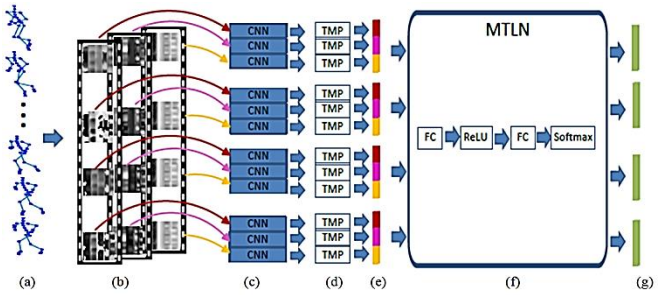

Figure 1. Architecture of the proposed method. Given a skeleton sequence (a), three clips (b) corresponding to the three channels of the cylindrical coordinates are generated. A deep pre-trained CNN model (c) and a temporal mean pooling (TMP) layer (d) are used to extract a compact representation from each frame of the clips (see Figure 2 for details). The output CNN representations of the three clips at the same time-step are link together, resulting four feature vectors (e). Every feature vector represents the temporal information of the skeleton sequence and a testing spatial association of the skeleton joints. The proposed MTLN (f) which includes a fully connected (FC) layer, a rectified linear unit (ReLU), another FC layer and a Soft max layer jointly processes the four feature vectors in parallel and outputs four sets of class scores (g), each corresponding to one task of classification using one feature vector. During preparation, the loss values of the four tasks are added up to define the loss value of the network used to update the network parameters. For testing, the class scores of the four tasks are averaged to generate the final Prediction of the action class.

The generated clips are then fed to a pre-trained CNN model and the proposed MTLN for robust feature learning and action recognition.

A. Clip Generation

Given a skeleton sequence, just the trajectory of the 3D Cartesian coordinates of the skeleton joints is provided. As mentioned in Section 1, the features extracted from the native 3D format (i.e., coordinates of joints) are sensitive to joint noise and temporal variations. This paper aims to transform the original skeleton sequence to a collection of clips consisting of images, which can be used to learn robust features using deep networks.

To transform a skeleton sequence to a video-based demonstration, intuitively, one could represent the substance of each frame of the skeleton sequence as an image, and then fuse all frames in a video. Though, this technique will outcome in a long video of which the temporal dynamics will be complicated to learn. In addition, each frame of the generated video will also be too sparse as the number of the skeleton joints is small.

In this paper, we propose to represent the temporal dynamics of the skeleton sequence in a frame image, and then use multiple frames to incorporate different spatial relationships between the joints. An benefit of this method is that for any skeleton sequence with an random number of frames, the generated clips contain the same number of frames. The robust and invariant temporal information of the original skeleton sequence could be captured with the powerful CNN representations erudite from each frame image.

Specifically, the time series of each joint of a skeleton sequence can be represented as three 1D feature columns corresponding to the three channels of the 3D Cartesian coordinates (x; y; z). To transform these time series of all joints to an image-based format, a simple way is to concatenate the 1D feature columns of all joints along the row dimension in a sequential order. A 2D array could thus be generated for each channel of the 3D coordinates. The 2D arrays could further be transformed to images by scaling their values. The drawback of this method is that it neglects the spatial association between the joints, which is a critical cue for action recognition as it describes a particular posture of a human.

To tackle this issue, instead of directly using the coordinates of each joint, this paper selects several

joints as reference joints. For each reference joint, a set of vectors can be derived by computing the difference of coordinates between the reference joint and the other joints. Each set of vectors reflects meticulous spatial relationships between the joints. In this paper, four joints are selected as the reference joints. The four reference joints are selected from four body parts, namely, the left shoulder, the right shoulder, the left hip and the right hip. The four joints are selected due to the fact that they are stable in most actions. They can thus reflect the motions of the other joints. Although the base of the spine is also stable, it is close to the left hip and the right hip. It is therefore discarded to avoid information redundancy. The four joints are respectively compared with other joints to generate four sets of vectors. The four sets of vectors combine different spatial relationships between the joints, providing useful spatial structural information of the skeleton joints.

More specifically, given a frame of a skeleton sequence, let the 3D coordinates of the skeleton joints be:

$$\Omega = \{q_i \in R^3 : i = 1, m\} \qquad (1)$$

Where m is the number of the skeleton joints, and $q_i=[x_i, y_i, z_i]$ represents the 3D coordinate of the i^{th} joint.

Let the reference joint be $q_0^k = [x_0^k\ y_0^k\ z_0^k] k = 1, \ldots .4$, and define

$$V_k \triangleq \{ q - q_0^k : q \epsilon \Omega , k = 1, \ldots ,4 \} \qquad (2)$$

Where V_k is the set of the vectors of the k^{th} reference joint in one frame.

The 3D Cartesian coordinates of each vector in V_k are further transformed to cylindrical coordinates. The cylindrical coordinates have been used to extract view-invariant motion features for action recognition [30]. Compared to the Cartesian coordinates, the cylindrical coordinates are more useful to analyze the motions as each human body utilizes pivotal joint movements to perform an action. Given a vector (*x*, *y*, *z*), the qualities are changed to (θ, ϕ, z) where θ = atan2(y/x), ϕ = $\sqrt{x^2 + y^2}$. For the k^{th} reference joint, all of the vectors in the set Vk are arranged in a chain. The three channels of the cylindrical coordinates of all vectors are separately concatenated into three rows, each corresponding to one channel of the cylindrical coordinates of all vectors.

Given a skeleton sequence with t frames, there are t sets of vectors for the k^{th} reference joint. The three rows of the t sets are separately concatenated along the row dimension, resulting three arrays D_θ^k ,D_ϕ^k ,D_z^k with R^{t*m}, m m is the number of vectors in each frame of the skeleton sequence. Each array can be transformed into a 2D gray image by scaling the values of the array between 0 and 255 using linear transformation. Thus for each channel of the cylindrical coordinates, the four reference joints generate four images, which are then combined in a clip. Consequently, three clips corresponding to the three channels $\theta, \emptyset, z$ are obtained.

Every frame of the generated clips describes the temporal dynamics of the entire frames of the skeleton sequence in one channel of the cylindrical coordinates. Particularly, the rows of the frame image communicate to the frames of the skeleton sequence, and the columns communicate to the vectors generated from the joints.

B. Clip Learning

The generated clips are different from the normal videos, i.e., there is no temporal order of the frames. For each clip, each frame includes one particular spatial relationship between the skeleton joints in one channel of the cylindrical coordinates. Different frames describe different spatial relationships and there exists intrinsic relationships among them. Therefore, instead of computing optical flow and learning the temporal structure of each clip to provide a video-level prediction, this paper proposes to extract a compact representation from each frame using a deep CNN. The three CNN features of the three clips at the alike time-step are link together in a feature vector, which represents the temporal information of the skeleton sequence and one particular spatial association between the skeleton joints in three channels of the cylindrical coordinates. Then the feature vectors of all time-steps are jointly processed in parallel using multi-task learning, thus to utilize their intrinsic relationships for action recognition.

C. Temporal Pooling of CNN Feature Maps

To learn the features of the generated clips, a deep CNN is firstly working to extract a compact

representation of each frame. Since each frame describes the temporal dynamics of the skeleton sequence, the spatial invariant CNN feature of every frame might therefore correspond to the strong temporal information of the skeleton sequence.

Given the generated clips, the CNN feature of every frame is extracted with the pre-trained VGG19 [25] model. The pre-trained CNN model is leveraged as a feature extractor due to the reality that the CNN features extracted by the models pre-trained with Image Net [22] are very powerful and have been efficiently applied in a number of cross-domain applications [3, 6, 21, 10]. In addition, current skeleton datasets are also too small or too noisy to correctly train a deep network. However the frames of the generated clips are not normal images, they might still be fed to the CNN model pre-trained with Image Net [22] for feature extraction. The similarity between a natural image and the generated frames is that both of them are matrices with some patterns. The CNN models trained on the large image dataset can be used as a feature extractor to extract representations of the patterns in matrices. The learned representations are generic and can be transferred to novel tasks from the original
tasks [34, 17].

The pre-trained VGG19 [25] model contains 5 sets of convolution layers conv1, conv2... conv5. Each set includes a stack of 2 or 4 convolution layers with the same kernel size. Totally there are 16 convolution layers and three fully connected layers in the network.

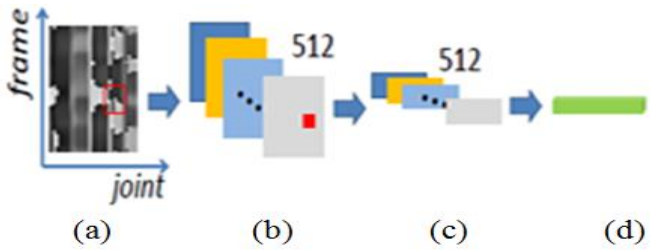

Figure 2. Temporal mean pooling of the CNN feature maps. (a) An input frame of the generated clips, for which the rows correspond to the different frames of the skeleton sequence and the columns correspond to the different vectors generated from the joints. (b)

Output feature maps of the conv5_1 layer. The size is 14 x14x 512. Each activation (shown in red) of the feature map is a feature correspond to the local region of the original image (shown with a red square). (c) Temporal features of all joints of the skeleton sequence, which are obtained by applying mean pooling to each feature map in the row (temporal) dimension. (d) Output feature, which is achieved by concatenating all the feature maps in (c).

Although deep neural networks are able to learn powerful and generic features which can be used in other novel domains, the features extracted from the different layers have different transferability. Particularly, the features in earlier layers are more generic, while in later layers; the features are more task-specific, which largely rely on the original classes and dataset. The features of the later layers are thus less suitable than those of the earlier layers to transfer to other domains [34, 17]. Therefore, this paper adopts a compact representation that is consequent from the activations of the convolution layer to exploit the temporal information of a skeleton sequence. The feature maps in the convolution layer have been successfully applied for action recognition and image retrieval [19, 20]. Specifically, the last 3 convolution layers and fully connected layers of the network are discarded. Each frame image of the three clips is scaled to 224x224, and is then duplicated three times to formulate a color image, so that it can be fed to the network. The output of the convolution layer conv5_1 is used as the representation of the input frame, which is a 3D tensor with size 14x14x512, i.e., 512 feature maps with size 14x14.

As mentioned in Section A, the rows of the generated frame correspond to different frames of a skeleton sequence. The dynamics of the row features of the generated image therefore represents the temporal evolution of the skeleton sequence. Meanwhile, the activations of each feature map in the conv5_1 layer are the local features corresponding to the local regions in the original input image [19]. The temporal information of the sequence can thus be extracted from the row features of the feature maps. More specifically, the feature maps are applied temporal mean pooling with kernel size 14 _ 1, i.e., the pooling is applied over the temporal, or row dimension, thus to generate a dense fusion representation from all temporal stages of the skeleton sequence.

Let the activation at the i^{th} push and the j^{th} segment of the k^{th} highlight guide be $x_{i,j}^{k}$. After temporal mean pooling, the output of the k^{th} feature map is given by:

$$y^k = [y_1^k \dots, y_j^k \dots, y_{14}^k] \quad (3)$$
$$y_j^k = \frac{1}{1}\sum_{i=0}^{14} \max(0, x_{i,j}^k)$$

The outputs of all feature maps (512) are concatenated to form a 7168D (14 x512 = 7168) feature vector, which represents the temporal dynamics of the skeleton sequence in one channel of the cylindrical coordinates.

D. Multi-Task Learning Network

As shown in Figure 1(e), the output 7168D features of the three clips at the same time-step are concatenated, generating four feature vectors. Each feature vector represents the temporal dynamics of the skeleton sequence and includes one particular spatial link between the joints in three channels of the cylindrical coordinates. The four feature vectors have intrinsic relations between each other. An Multi task learning network is then proposed to jointly process the four feature vectors to utilize their intrinsic relationships for action recognition. The classification of each feature vector is treated as a separate task with the same classification label of the skeleton sequence.

The architecture of the network is shown in Figure 1(f). It includes two fully connected (FC) layers and a Softmax layer. Between the two FC layers there is a rectified linear unit (ReLU) [18] to introduce an additional non-linearity. Given the four features as inputs, the MTLN generates four frame-level predictions, each corresponding to one task.

During training, the class scores of each task are used to compute a loss value. Then the loss values of all tasks are summed up to generate the final loss of the network used to update the network parameters. During testing, the class scores of all tasks are averaged to form the final prediction of the action class. The loss value of the k^{th} task (k = 1,….,4) is given by Equation 4.

$$L_k(Z_k\ Y) = \sum_{i=1}^{m} y_i\ \left(-\log\left(\frac{exp_{z_{ki}}}{\sum_{j=1}^{m} exp_{z_{kj}}}\right)\right) \quad (4)$$

$$= \sum_{i=1}^{m} y_i\ \left(\log \sum_{j=1}^{m} exp_{z_{kj}}\right) - zk_i$$

Where z_k is the vector fed to the Softmax layer generated from the k^{th} input feature, m is the amount of action classes and y_i is the ground-truth label for class i. The final loss value of the network is computed as the sum of the four individual losses, as shown below in Equation 5:

$$L(Z, Y) = \sum_{k=1}^{4} l_k\ (Z_k\ Y) \quad (5)$$

Where Z= [$Z_{1\dots,}$ Z4].

IV. Conclusion

Finally In this paper, we have proposed to transform a skeleton sequence to three video clips for robust feature learning and action recognition. We proposed to use a pre-trained CNN model followed by a temporal pooling layer to extract a compact representation of each frame. The CNN features of the three clips at the same time-step are concatenated in a single feature vector, which describes the temporal information of the total skeleton sequence and one particular spatial connection between the joints. We then propose an MTLN to jointly learn the feature vectors at all the time steps in similar, which utilizes their intrinsic relationships and improves the performance for action recognition. We have tested the proposed method on three datasets, including NTU RGB+D dataset, SBU kindest interaction dataset and CMU dataset. Experimental results have shown the effectiveness of the proposed new representation and feature learning method.

V. REFERENCES

[1]. R. Caruana. Multitask learning. In Learning to find out, pages 95-133. Springer, 1998.

[2]. CMU. CMU graphics lab motion capture folder. In http://mocap.cs.cmu.edu/. 2013.

[3]. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional formation element for generic visual detection. In International Conference on Machine Learning , pages 647-655, 2014.

[4]. Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton base act detection. In IEEE Conference on PC Vision and Pattern detection , pages 1110-1118, 2015.

[5]. G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads: person act judgment with joint quadruples. In International Conference on Pattern identification , pages 4513-4518, 2014.

[6]. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for correct object detection and semantic Parts. In IEEE Conference on PC Vision and Pattern detection , pages 580-587, 2014.

[7]. A. Graves. Neural networks. In Supervised approved Labeling with Recurrent Neural Networks, pages 15-35. Springer, 2012.

[8]. A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing, pages 6645-6649. IEEE, 2013.

[9]. F. Han, B. Reily, W. Hoff, and H. Zhang. space-time demonstration of people base on 3d skeletal data: a review. arXiv preprint arXiv:1601.01006, 2016.

[10]. X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Match net: Unifying feature and metric learning for area based matching. In IEEE Conference on processor Vision and Pattern identification, pages 3279-3286, 2015.

[11]. J.F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for RGB-D activity detection. In IEEE Conference on Computer Vision and Pattern detection, pages 5344-5352, 2015.

[12]. M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action discovery using a temporal hierarchy of covariance descriptors on 3d joint points In IJCAI, volume 13, pages 2466-2472, 2013.

[13]. Y. Ji, G. Ye, and H. Cheng. Interactive body part contrast mining for individual interaction detection. In IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pages 1-6. IEEE, 2014.

[14]. P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for act detection from 3d skeletons. arXiv preprint arXiv:1604.00239, 2016.

[15]. W. Li, L. Wen, M. Choo Chuah, and S. Lyu. Category-blind individual action recognition: A useful recognition system. In IEEE International Conference on Computer Vision (ICCV), pages 4444-4452, 2015.

[16]. J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal LSTM with belief gates for 3D human being action detection. In European Conference on PC Vision (ECCV), pages 816-833. Springer, 2016.

[17]. M. Long and J. Wang. Learning transferable features with deep adaptation networks. CoRR, abs/1502.02791, 1:2, 2015.

[18]. V. Nair and G. E. Hinton. Rectified linear units progress restricted boltz mann tools. In International Conference on Machine knowledge, pages 807-814, 2010.

[19]. X. Peng and C. Schmid. Encoding feature maps of cnns for act detection. 2015.

[20]. F. Radenovi´c, G. Tolias, and O. Chum. Cnn image recovery learns from bow: Unsupervised fine-tuning with solid examples. arXiv preprint arXiv:1604.02426, 2016.

[21]. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for detection. In IEEE Conference on PC Vision and Pattern detection Workshops ,pages 806-813, 2014.

[22]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet huge scale visual detection challenge, International Journal of Computer Vision, 115(3):211-252, 2015.

[23]. A. Savitzky and M. J. Golay. smooth and partition of data by simplify slightest squares procedures. Analytical chemistry, 36(8):1627-1639, 1964.

[24]. A. Shahroudy, J. Liu, T.-T. Ng, and G.Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In IEEE Conference on PC Vision and Pattern detection , June 2016.

[25]. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image identification. arXiv preprint arXiv:1409.1556, 2014.

[26]. C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual person motion detection. Computer Vision and Image Understanding, 104(2):210-220, 2006.

[27]. A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In ACM International Conference on Multimedia, pages 689-692, 2015.

[28]. R. Vemulapalli, F. Arrate, and R. Chellappa. Person act detection by representing 3d skeletons as points in a lie group. In IEEE Conference on Computer Vision and Pattern identification ,pages 588-595, 2014.

[29]. J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining action let ensemble for action discovery with depth cameras. In IEEE Conference on Computer Vision and Pattern detection , pages 1290-1297, 2012.
[30]. D. Weinland, R. Ronfard, and E. Boyer. Free view point action recognition using movement history volumes. Computer vision and image perceptive, 104(2):249-257, 2006.
[31]. D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In IEEE Conference on Computer Vision and Pattern Recognition , pages 724-731, 2014.
[32]. L. Xia, C.C. Chen, and J. Aggarwal. View invariant human action recognition with histograms of 3D joint. In IEEE Conference on Computer Vision and Pattern Recognition Workshops , pages 20-27, 2012.
[33]. X. Yang and Y. L. Tian. Eigen joints-based action recognition with naive-bayes-nearest-neighbor. In IEEE Computer Society Conference on Computer Vision and Pattern identification Workshops , pages 14-19, 2012.
[34]. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How moveable are features in deep neural networks? In Advances in neural information processing systems, pages 3320-3328, 2014.
[35]. K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body pose features and multiple occurrence learning. In IEEE Conference on Computer Vision and Pattern identification Workshops , pages 28-35, 2012.
[36]. W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for bones based action identification using regularized deep lstm networks. In AAAI Conference on Artificial Intelligence, 2016.