Training CNNs for 3-D Sign Language Recognition With Color Texture Coded Joint Angular Displacement Maps

E. Kiran Kumar, *Student Member, IEEE*, P. V. V. Kishore, *Senior Member, IEEE*, A. S. C. S. Sastry, *Member, IEEE*, M. Teja Kiran Kumar, *Student Member, IEEE*, and D. Anil Kumar, *Student Member, IEEE*

***Abstract*—Convolutional neural networks (CNNs) can be remarkably effective for recognizing two-dimensional and three-dimensional (3-D) actions. To further explore the potential of CNNs, we applied them in the recognition of 3-D motion-captured sign language (SL). The sign's 3-D spatio-temporal information of each sign was interpreted using joint angular displacement maps (JADMs), which encode the sign as a color texture image; JADMs were calculated for all joint pairs. Multiple CNN layers then capitalized on the differences between these images and identify discriminative spatio-temporal features. We then compared the performance of our proposed model against those of the state-of-the-art baseline models by using our own 3-D SL dataset and two other benchmark action datasets, namely, HDM05 and CMU.**

***Index Terms*—Convolutional neural network, joint angular displacement map, three-dimensional (3-D) sign language recognition.**

I. Introduction

SIGN language recognition (SLR) is a complicated problem that involves attempting to accurately translate visual signals into text/voice in real time. With the release of low-cost real-time depth sensors, such as the Kinect and Leap Motion sensors, the focus of SLR has shifted from two-dimensional (2-D) to three-dimensional (3-D) RGB–D (red, green, blue–depth) approaches. The RGB data and depth information are processed separately to build feature vectors that are then classified using a pattern classifier, such as a hidden Markov model, support vector machine, or artificial neural network [1], [2].

In one study, Leap Motion sensors were used to capture real hand movements, which were then projected as 3-D animated hand models for the recognition phase [3]. Several recent papers have explored 3-D SLR [4], [5] using these sensors. The 3-D data produced by Kinect sensors consists of hand trajectories [5], orientations, and velocities [6] taken from individual depth images. Features such as 3-D body joint locations [7] and finger–earth mover's distances [8] could be used to classify signs by using such data. Improved accuracy and recognition rates have been reported for both Kinect and Leap Motion sensors than compared with those for 2-D models.

Encouraged by the progress made by recent studies on recognizing human actions with convolutional neural networks (CNNs) [9]–[12], we propose the application of a CNN to SLR on the basis of 3-D motion capture (mocap) Indian sign language (SL) data [13]. However, previous studies have found, however, that there can be ambiguity when encoding 3-D skeletal data into color images. For example, the joint distance maps (JDMs) between skeletal joints have been encoded into color texture images for use with CNNs [14]. Similarly, reference JDMs have been used to classify images in the RGB–D database of Nanyang Technological University by using long short-term memory networks and CNNs [15]. These encodings have resulted in high accuracies when using Kinect skeletal data or 3-D mocap data for action recognition, and the authors show that a multichannel CNN trained on multiview data can achieve reasonably good recognition rates.

However, JDM-based approaches to SLR have been found to have high misclassification rates due to small interfinger movements and large hand movements against backgrounds where the head, chest, and face were static. The small distance changes for the fingers have little impact on the pixel colors, thus leading to minimal interclass variation. Thus, this study investigates an alternative approach of encoding joint angular displacement maps (JADMs) into color texture images for use in CNN-based recognition. We believe this approach will allow us to generate features that can better discriminate between different sign classes.

JADMs use both joint angular measurements and JDMs to encode 3-D mocap or skeletal data into color texture images. The proposed method is based on [14]–[16] and [17]. The joints' 3-D coordinates (x, y, z) [15] or the relative joint positions in the x–y, y–z, and z–x planes [16] can be encoded in the RGB planes of a color image. However, both of these approaches have

Manuscript received December 9, 2017; revised March 3, 2018; accepted March 15, 2018. Date of publication March 19, 2018; date of current version April 4, 2018. This work was supported in part by the research project scheme titled "Visual–Verbal Machine Interpreter Fostering Hearing Impaired and Elderly" by the "Technology Interventions for Disabled and Elderly" program of the Department of Science and Technology, SEED Division, Govt. of India, Ministry of Science and Technology under Grant SEED/TIDE/013/2014(G). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mehdi Moradi. *(Corresponding author: P.V.V. Kishore.)*

The authors are with the Biomechanics and Vision Computing Research Center, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Guntur 522502, India (e-mail: kiraneepuri@kluniversity.in; pvvkishore@kluniversity.in; ascssastry@kluniversity.in; mtejakiran@kluniversity.in; danilmurali@kluniversity.in).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2018.2817179

view-invariance issues, which can be overcome with JDMs [14]. The colors in JDM images encode the joint distances for 3-D action sequences, but an attempt to use them with a single-channel CNN required longer training and still failed a view-invariance test. An alternative proposal used four-channel CNNs [14], which solved the view-invariance problem but still needed longer training times. The four channels required four view-invariant images to be generated from a single 3-D video.

By combining JDMs with angular information, JADMs are able to deal with these issues. We demonstrate the advantages of our JADM-based encoding by training a single-channel CNN and testing its performance, view invariance, and training time by using both our own 3-D mocap SL dataset and two other publicly available datasets, namely HDM05 [18] and CMU [19]. The results show that the SL classification performance of our model is better than those of the state-of-the-art baseline methods.

II. Deep Learning with JADMs

The problem tackled here can be stated as follows. On the basis of labeled 3-D mocap sign videos, we derived color texture JADMs for each video and used them to train a CNN to predict the sign labels. The 3-D SL gesture data consists of human motions that are both spatially and temporally irregular.

The proposed method involves the following three steps:

1) Compute JADMs from the 3-D data.
2) Encode the JADMs into RGB images.
3) Use the images to train and test a CNN.

Fig. 1 shows the CNN architecture used. The previous approaches [14]–[17] are based on projecting the 3-D skeletal joint positions onto three orthogonal planes, derived from a real-world coordinate system centered on the camera, thus resulting in so-called paired JDMs. These are then used to train a complex multichannel CNN architecture. Compared with this method, our approach is simpler in the following aspects:

1) It uses JADMs for all pairs of joints.
2) The information is encoded using only the color intensity, rather than as RGB data.
3) Only a simple single-channel CNN is needed.

The use of JADMs as input to the CNN results in better recognition and shorter training times, and allows the system to be view invariant.

Each frame of a sequence involving a 3-D skeleton with J joints will include $\frac{J\times(J-1)}{2}$ unique joint pairs. Each joint can be represented as a position vector in 3-D space: $(x_i, y_i, z_i) \in R^{3\times J}\ \forall\, i = 1$ to J. The JAD for a frame is a vector with $\frac{J\times(J-1)}{2}$ entries, one for each joint pair. For a T-frame sequence, the overall JAD is a matrix of size $(\frac{J\times(J-1)}{2}) \times T$. This characterizes the spatio-temporal changes in the JADs and is encoded in a color texture image of the same size. Given that different videos include different numbers of frames, we apply binary interpolation to the RGB planes of the encoded images to increase the number of frames to the same value, T_C, changing the image size to $(\frac{J\times(J-1)}{2}) \times T_C \times 3$. The following section describes the process for extracting JADMs from the joint locations of a 3-D skeleton in detail.

A. Creating JADMs

In 3-D space, the position p_i of the ith joint (of *J*) can be represented using 3-D coordinates as $p_i(x_i, y_i, z_i) \in R^{3\times J}\ \forall\, i = 1$ to J. For the complete set of J joints, we then have the combined position vector $p_i = [p_1, p_2, ...p_J]$. Thus, a *T*-frame sign (or other sequence of actions) A can be expressed as $A = \{P_1, P_2, ..., P_T\} \in R^{J\times 3\times T}$. In general, the l_2 distance norm d_{ij}^t between the ith and jth joint during the tth frame of A is formulated as [14]

$$d_{ij}^t = \left\| P_i^t - P_j^t \right\|_2 . \tag{1}$$

Here, we describe the relationships between joints by using the JAD d_{ij}^θ between the ith and jth joint during the tth frame of A. This is formulated as

$$d_{\measuredangle ij}^t = d_{ij}^t \cos\left(\theta_{ij}^t\right) \tag{2}$$

where θ_{ij}^t is the angle between the ith and jth joint during the tth frame. Calculating this angle requires a minimum of three joint projections. By using markers J_1, J_2, and J_3, we can form two projection vectors, namely, $\overrightarrow{P_{12}} = d(J_1, J_2) \in R^3$ and $\overrightarrow{P_{23}} = d(J_2 J_3) \in R^3$, where $d(J_i, J_j)$ is the distance between markers J_i and J_j. On the basis of these vectors, we can measure the angle at J_2 as

$$\theta_{J_2} = \cos^{-1} \frac{\overrightarrow{P_{12}}\overrightarrow{P_{23}}}{\sqrt{\overrightarrow{P_{12}}}\sqrt{\overrightarrow{P_{23}}}}. \tag{3}$$

The JAD $d_{\measuredangle ij}^T$ for the ith and jth joint over all frames in the sequence is a row vector consisting of the JADs for each frame. Therefore, the rows of the JAD matrix can be expressed as

$$d_{\measuredangle ij}^T = \left[d_{\measuredangle ij}^1, d_{\measuredangle ij}^2, d_{\measuredangle ij}^3, ..., d_{\measuredangle ij}^T \right]. \tag{4}$$

The JAD matrix encapsulates three types of information about the motion, namely, the joint distances, angles, and time evolution. Finally, it is encoded in an RGB image in JPEG format for input to the CNN.

B. Encoding JADMs into RGB Images

Unlike previous studies [14], [15], we encode the JAD matrix into an image simply, using a standard mapping procedure [20] with the "jet" color map. Combining the three RGB color planes into one creates a JAD image consisting solely of intensity values. Previous approaches have encoded distance maps into color images [14], [15], but these are influenced by the dimensions of the subject, thus resulting in increased numbers of misclassifications. Therefore, such approaches have to normalize the dimensional variations of the subjects either in the skeletal model or the encoded images. In the present study, we used the angles between joints to compensate for variations in their distances, thus making our approach immune to dimensionality variations. Fig. 2 shows how the JADM for a 3-D sign video is encoded.

The previous studies [14], [15] represented each frame by using four color-coded images, in the *x–y*, *y–z*, *z–x*, and *xyz* planes, to improve the accuracy of the CNN's. By contrast, with our JADM approach, only a single RGB image is needed to train and test the CNN classifier.

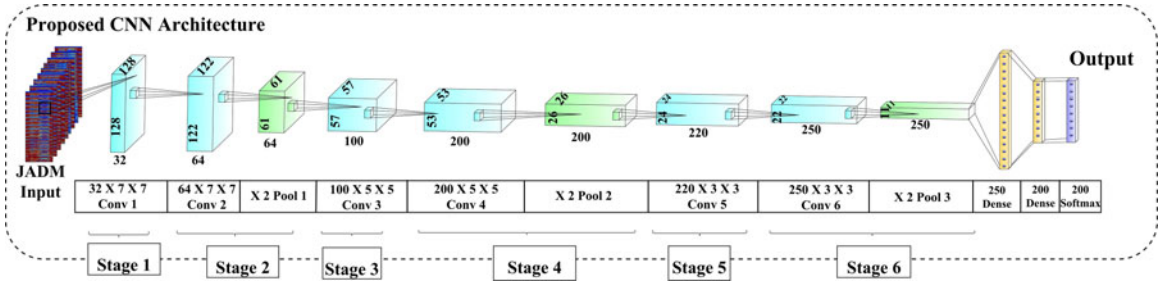

Fig. 1. Proposed CNN architecture.

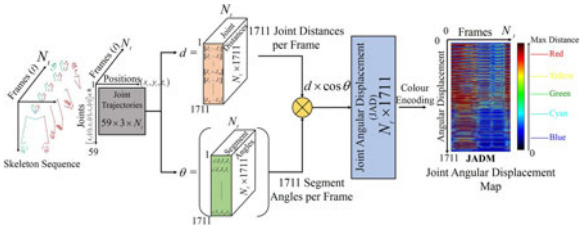

Fig. 2. JADM encoding process.

III. Experimental Method

A. CNN Architecture

The proposed 3-D SignNet CNN is inspired by the VGG architecture introduced by Simonyan and Zisserman [21], which is a very deep CNN model that demonstrated the state-of-the-art classification and the localization accuracy in the ImageNet Large Scale Visual Recognition Challenge in 2014. VGG is a deep CNN with 16–19 weighted convolutional layers with small (3×3) window sizes, this architecture is same as that in the CNNs originally proposed by Ciresan *et al.* [22] and Jeffrey Dean *et al.* [23]. Our 3-D SignNet architecture (Fig. 1) is similar, but the depth is limited to six weighted layers and two fully connected layers.

The simulations were implemented using Python 3.6 with the help of the Keras and TensorFlow libraries, with substantial adjustments made during testing and training. They were then executed on a 12-GB GPU (NVIDIA Tesla K20M 2×6 GB) linked to a six-node high-performance computer.

B. Datasets

All 3-D sign skeletons were represented using 57 human upper body joints. The 3-D template consisted of 57 markers: 18 each for the left and right hands, 2 for the shoulders, 1 for the chest, 2 for the arms, 12 for the face, and 4 for the head. These were listed by all the datasets in a fixed order, starting with the head before moving on to the face, chest, right shoulder, right hand, right fingers, left shoulder, left hand, and left fingers.

We created a 3-D mocap SL dataset consisting of 200 Indian SL signs, which we captured from ten different signers by using a 9-cam 3-D mocap system. Each signer repeated each sign ten times, with variations in hand speed and trajectory, changes the in part of the face or head that was being focused on, and variations in the times each pose was held. This resulted in a total of 20 000 3-D sign videos, divided into 200 classes with 100 signs per class. For each class, 50 randomly chosen videos were used for training and the remaining 50 were used for testing.

Furthermore, we used the HDM05 [18] and CMU [19] action datasets to validate both the proposed image creation process and the single-channel CNN. From the HDM05 action dataset, we used a total of 1500 actions from 23 classes. From the CMU action dataset, we used a total of 700 actions from 70 classes by ten different actors.

C. Training

We trained the network by using these three datasets with the same layer definitions and filter configurations for each run, as well as the training algorithm presented by Dean *et al.* [23]. This involved optimizing a multinomial logistic regression objective function by means of mini-batch gradient descent with a momentum of 0.9. The weights in each layer were initialized randomly by using a Gaussian distribution function with a mean of zero and a variance of 0.01.

All the JADM images were resized to 128×128 by using bicubic interpolation, and the l_2 penalization multiplier was set to 0.0002 to regularize the weight decay during training for this image size. Dropout regularization was used for the first four layers with a rate of 0.5.

The learning rate was initially set to 0.02 and decreased by a factor of 10 when the validation accuracy became constant. For the 10 000-sample sign dataset, this meant that the learning rate was decreased three times from 0.02 to 0.005 and then to 0.001. The training stopped after 100 epochs (31.25 k iterations), and the learning rate decreased after every 48 epochs (15 k iterations) thereafter. For the HDM05 dataset (750 training samples), training stopped after 48 epochs (1.125 k iterations), and the learning rate decreased after every 22 epochs (0.517 k iterations). For the CMU dataset (350 training samples), training stopped after 36 epochs (0.394 k iterations), and the learning rate decreased after every 20 epochs (0.219 k iterations). Fewer epochs were required in these cases because of the use of medium-scale images. Increasing the image size to 224×224 brought little or no improvements in the predicted labels.

D. Testing

The trained CNNs were used to classify the actions in each testing dataset by using 128×128 encoded images as input and outputting class score vectors with an entry for each class. To speed up the testing process, we rescaled the images to

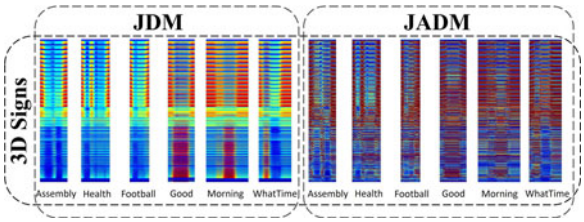

Fig. 3. Color texture image comparison for JDMs [14] and the proposed JADMs.

TABLE I
PERFORMANCE OF JDM VS. JADM FOR THE SIGN, CMU, AND HDM05 DATASETS

Dataset		Recognition (%)		Precision (%)		Recall (%)	
	JDM	JADM	JDM	JADM	JDM	JADM	
Front view	Sign	75.66	89.93	75.42	89.43	80.61	91.21
	HDM05	86.24	91.12	84.21	88.1	87.44	93.54
	CMU	85.99	90.33	84.61	88.23	88.24	93.46
Cross view	Sign	71.51	88.69	70.26	87.67	75.81	91.44
	HDM05	84.93	89.15	82.54	89.22	88.66	92.14
	CMU	82.87	88.67	81.31	86.98	87.24	92.71
Cross subject	Sign	73.96	88.59	74.24	85.92	82.57	90.01
	HDM05	84.53	87.92	83.92	85.61	86.96	90.33
	CMU	83.14	87.27	82.57	84.25	88.54	92.25

100×100 because this size produced the same results as the 128×128 images for a test dataset.

The results were evaluated in terms of the recognition, precision, and recall metrics on the test datasets. We compared the performance of our approach with that of a range of the state-of-the-art methods by using data taken from relevant papers.

IV. RESULTS AND ANALYSIS

We tested our proposed JADM-based encoding method by using three datasets: our own SL dataset and two publicly available datasets: HMD05 and CMU. We built six CNNs by using the proposed architecture and trained them by using images via JADM and JDM encoding [14]. Fig. 3 shows the example JDMs and JADMs from the sign dataset.

First, we compared the proposed JADM + CNN architecture with a similar but JDM-based architecture (calculating the JDMs as in Li *et al.* [14]), for the SL dataset. By using JADMs, training took 198 epochs, compared with 350 epochs for JDMs. However, we only achieved the same recognition rate with JDMs after 502 epochs. Table I shows the average recognition, precision, and recall values for the JADM and JDM images on the three datasets using front, cross, and cross subject views. The cross and cross subject view data was generated by rotating the subject skeletons by $+45^\circ$ and -45° horizontally and vertically, respectively.

As Table I shows, JADMs performed better than JDMs on all three datasets, which can be attributed to their inclusion of joint angular information.

TABLE II
PERFORMANCE COMPARISON FOR THE PROPOSED AND BASELINE METHODS ON THE CMU DATASET

Method	Recognition (%)		Precision (%)		Recall (%)	
	Cross subject	Cross view	Cross subject	Cross view	Cross subject	Cross view
Hierarchical RNN [24]	71.73	75.02	70.32	74.19	76.95	80.99
Deep LSTM [25]	72.54	79.53	72.02	78.45	78.36	83.26
Deep LSTM+ co-occurrence [25]	72.81	79.63	73.13	79.23	78.49	83.41
SkeletonNet [10]	80.46	84.83	80.19	83.84	85.01	86.82
JDM + CNN	83.14	82.87	82.57	81.31	88.54	87.24
JADM + CNN	87.27	88.67	84.25	86.98	92.25	92.71

TABLE III
PERFORMANCE COMPARISON FOR THE PROPOSED AND BASELINE METHODS ON THE HDM05 DATASET

Method	Recognition (%)		Precision (%)		Recall (%)	
	Cross subject	Cross view	Cross subject	Cross view	Cross subject	Cross view
Multilayer Perceptron [26]	81.52	84.52	80.14	84.25	84.42	87.76
Hierarchical RNN [24]	83.98	84.98	83.05	85.09	86.65	87.81
Deep LSTM [25]	84.24	85.26	83.98	85.13	86.89	88.32
SPDNet [27]	61.45	65.63	62.03	66.19	69.59	72.53
SE [28]	70.26	72.49	71.48	73.14	75.64	79.63
SO [29]	71.31	75.68	71.96	74.59	76.58	79.54
LieNet-2Blocks [30]	75.78	79.89	76.04	80.21	80.85	82.56
JDM+CNN	84.53	84.93	83.92	82.54	86.96	88.66
JADM+CNN	**87.92**	**89.15**	**85.61**	**89.22**	**90.33**	**92.14**

We also compared the performance of the proposed method with that of five other deep learning architectures, namely, a hierarchical RNN [24], a deep LSTM (with and without cooccurrence) [25], SkeletonNet [10], and our method with JDMs (JDM+CNN). Table II shows their respective results on the CMU dataset, for the cross and cross subject views. These results show that our proposed CNN architecture performed better in all cases because JADMs are both view and subject invariant.

A similar analysis was performed using HDM05 dataset, and the results are shown in Table III. The proposed method achieved better performance than the state-of-the-art baseline methods.

V. CONCLUSION

In this study, we proposed the use of JADMs for representing the spatio-temporal information in 3-D mocap videos. Unlike JDMs, these model local information by using the distances and angles between joint pairs. Furthermore, we have proposed CNN architecture for classifying the encoded images. We also evaluated the proposed method by comparing its performance with those of the state-of-the-art baseline methods by using our own SL dataset, as well as the CMU and HDM05 action datasets. The results show that our JADM encoding generates unique representations of 3-D mocap data and can be used for the machine translation of 3-D Indian SL with deep learning.

REFERENCES

[1] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2015. [Online]. Available: https://doi.org/10.1109

[2] C. Sun, T. Zhang, B.-K. Bao, C. Xu, and T. Mei, "Discriminative exemplar coding for sign language recognition with kinect," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1418–1428, Oct. 2013. [Online]. Available: https://doi.org/10.1109

[3] C.-H. Chuan, E. Regina, and C. Guardino, "American sign language recognition using Leap Motion sensor," in *Proc. 13th Int. Conf. Mach. Learn. Appl.*, IEEE, Dec. 2014. [Online]. Available: https://doi.org/10.1109

[4] S. G. M. Almeida, F. G. Guimarães, and J. A. Ramírez, "Feature extraction in Brazilian sign language recognition based on phonological structure and using RGB-d sensors," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7259–7271, Nov. 2014. [Online]. Available: https://doi.org/10.1016

[5] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji, "Feature learning based on SAE–PCA network for human gesture recognition in RGBD images," *Neurocomput.*, vol. 151, pp. 565–573, Mar. 2015. [Online]. Available: https://doi.org/10.1016

[6] L. Geng, X. Ma, H. Wang, J. Gu, and Y. Li, "Chinese sign language recognition with 3-D hand motion trajectories and depth images," in *Proc. 11th World Congr. Intell. Control Autom.*, IEEE, Jun. 2014. [Online]. Available: https://doi.org/10.1109

[7] W. Nai, Y. Liu, D. Rempel, and Y. Wang, "Fast hand posture classification using depth features extracted from random line segments," *Pattern Recog.*, vol. 65, pp. 1–10, May 2017. [Online]. Available: https://doi.org/10.1016

[8] Z. Zhang and A. V. Kurakin, "Dynamic hand gesture recognition using depth data," US Patent Appl. 15 334 269, Feb. 16, 2017.

[9] T. Qi, Y. Xu, Y. Quan, Y. Wang, and H. Ling, "Image-based action recognition using hint-enhanced deep neural networks," *Neurocomput.*, vol. 267, pp. 475–488, Dec. 2017. [Online]. Available: https://doi.org/10.1016

[10] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "SkeletonNet: Mining deep part features for 3-D action recognition," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 731–735, Jun. 2017. [Online]. Available: https://doi.org/10.1109

[11] S. Ji, W. Xu, M. Yang, and K. Yu, "3-D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013. [Online]. Available: https://doi.org/10.1109

[12] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep CNNs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 209–226, Feb. 2017. [Online]. Available: https://doi.org/10.1109

[13] P. Kishore, D. Kumar, A. Sastry, and E. Kumar, "Motionlets matching with adaptive kernels for 3-D Indian sign language recognition," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3327–3337, Apr. 2018. [Online]. Available: https://doi.org/10.1109

[14] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 624–628, May 2017. [Online]. Available: https://doi.org/10.1109

[15] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using LSTM and CNN," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2017. [Online]. Available: https://doi.org/10.1109

[16] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3-D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jul. 2017. [Online]. Available: https://doi.org/10.1109

[17] M. Liu, C. Chen, F. Meng, and H. Liu, "3-D action recognition using multitemporal skeleton visualization," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2017. [Online]. Available: https://doi.org/10.1109

[18] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Tech. Rep. CG-2007-2, Universität Bonn, Jun. 2007.

[19] CMU, "Cmu graphics lab motion capture database," 2013. [Online]. Available: http://mocap.cs.cmu.edu/

[20] P. Wang, W. Li, Z. Gao, C. Tang, J. Zhang, and P. Ogunbona, "ConvNets-based action recognition from depth maps through virtual cameras and pseudocoloring," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015. [Online]. Available: https://doi.org/10.1145

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[22] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. 20nd Int. Joint Conf. Artif. Intell.,*vol. 2, AAAI Press, 2011, pp. 1237–1242. [Online]. Available: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-210

[23] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. 25th Int. Conf. Neural Inform. Process. Syst.,*vol. 1, USA: Curran Associates Inc., 2012, pp. 1223–1231. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999271

[24] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015. [Online]. Available: https://doi.org/10.1109

[25] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," *CoRR*, vol. abs/1603.07772, 2016. [Online]. Available: http://arxiv.org/abs/1603.07772

[26] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," in *Proc. 9th Int. Conf. Comput. Vis. Theory Appl.* SCITEPRESS – Science and Technology Publications, 2014. [Online]. Available: https://doi.org/10.5220

[27] Z. Huang and L. V. Gool, "A Riemannian network for SPD matrix learning," *CoRR*, vol. abs/1608.04233, 2016. [Online]. Available: http://arxiv.org/abs/1608.04233

[28] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3-D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*., Jun. 2014. [Online]. Available: https://doi.org/10.1109

[29] R. Vemulapalli and R. Chellappa, "Rolling rotations for recognizing human actions from 3-D skeletal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.,* Jun. 2016. [Online]. Available: https://doi.org/10.1109

[30] Z. Huang, C. Wan, T. Probst, and L. V. Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jul. 2017. [Online]. Available: https://doi.org/10.1109