Make Skeleton-based Action Recognition Model Smaller, Faster and Better

Conference Paper · December 2019

DOI: 10.1145/3338533.3366569

4 authors, including:

Fan Yang
Nara Institute of Science and Technology
20 PUBLICATIONS **195** CITATIONS
SEE PROFILE

Sakriani Sakti
Nara Institute of Science and Technology
243 PUBLICATIONS **1,901** CITATIONS
SEE PROFILE

Satoshi Nakamura
Nara Institute of Science and Technology
814 PUBLICATIONS **7,634** CITATIONS
SEE PROFILE

Make Skeleton-based Action Recognition Model Smaller, Faster and Better

Fan Yang
Nara Institute of Science and Technology, Japan
RIKEN, Center for Advanced Intelligence Project, Japan
yang.fan.xv6@is.naist.jp

Yang Wu*
Kyoto University, Japan
wu.yang.8c@kyoto-u.ac.jp

Sakriani Sakti
Nara Institute of Science and Technology, Japan
RIKEN, Center for Advanced Intelligence Project, Japan
ssakti@is.naist.jp

Satoshi Nakamura
Nara Institute of Science and Technology, Japan
RIKEN, Center for Advanced Intelligence Project, Japan
s-nakamura@is.naist.jp

ABSTRACT

Although skeleton-based action recognition has achieved great success in recent years, most of the existing methods may suffer from a large model size and slow execution speed. To alleviate this issue, we analyze skeleton sequence properties to propose a Double-feature Double-motion Network (DD-Net) for skeleton-based action recognition. By using a lightweight network structure (*i.e.*, 0.15 million parameters), DD-Net can reach a super fast speed, as 3,500 FPS on an ordinary GPU (*e.g.*, GTX 1080Ti), or, 2,000 FPS on an ordinary CPU (*e.g.*, Intel E5-2620). By employing robust features, DD-Net achieves state-of-the-art performance on our experiment datasets: SHREC (*i.e.*, hand actions) and JHMDB (*i.e.*, body actions). Our code is on https://github.com/fandulu/DD-Net.

KEYWORDS

Skeleton-based Action Recognition, Body Actions, Hand Gestures

ACM Reference Format:
Fan Yang, Yang Wu, Sakriani Sakti, and Satoshi Nakamura. 2019. Make Skeleton-based Action Recognition Model Smaller, Faster and Better. In *ACM Multimedia Asia (MMAsia '19), December 15–18, 2019, Beijing, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3338533.3366569

1 INTRODUCTION

Skeleton-based action recognition has been widely used in multimedia applications, such as human-computer interaction [27], human behavior understanding [32] and medical assistive applications [3]. However, most of the existing methods may suffer from a large model size and slow execution speed [7, 9, 10, 14, 39].

In real applications, a desirable skeleton-based action recognition model should run efficiently by using a few parameters, and, also be adaptable to various application scenarios (*e.g.*, hand/body, 2D/3D skeleton, and actions related/unrelated to global trajectories). To

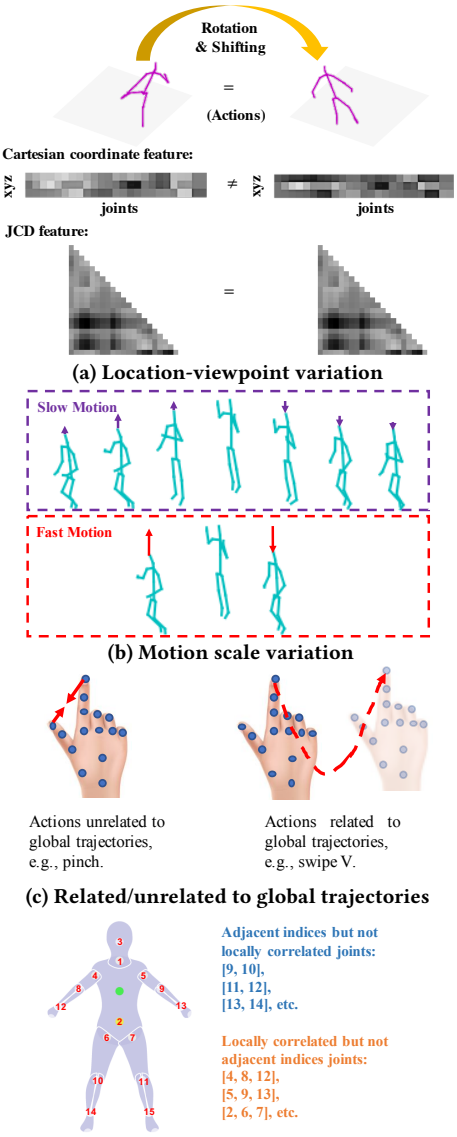

Figure 1: Concerned skeleton sequence properties.

*Corresponding author.

MMAsia '19, December 15–18, 2019, Beijing, China

ACM ISBN 978-1-4503-6841-4/19/12…$15.00
https://doi.org/10.1145/3338533.3366569

achieve this goal, we investigate skeleton sequence properties to propose a lightweight Double-feature Double-motion Network (DD-Net), which is equipped with a Joint Collection Distances (JCD) feature and a two-scale global motion feature.

More specifically, we conduct research on four types of skeleton sequence properties (see Fig. 1): (a) location-viewpoint variation, (b) motion scale variation, (c) related/unrelated to global trajectories, (4) uncorrelated joint indices. To address challenges caused by these properties, previous works may prone to propose complicated neural network models, which end up with large model size.

In contrast, we address these challenges by simplifying both the input feature and the network structure. Our JCD feature contains the location-viewpoint invariant information of skeleton sequences. Compared with other similar features, it can be easily computed and includes fewer elements. Since global motions cannot be incorporated into a location-viewpoint invariant feature, we introduce a two-scale global motion feature to improve the generalization of DD-Net. Besides, its two-scale structure makes it robust to the motion scale variance. Through an embedding process, DD-Net can automatically learn the proper correlation of joints, which is hard to be predefined by joint indices.

Compared to methods relying on complicated model structures, DD-Net provides higher action recognition accuracy and demonstrates its generalization on our experiential datasets. With its efficiency both in terms of computational complexity and the number of parameters, DD-Net is sufficient to be applied in real applications.

2 RELATED WORKS

Nowadays, with the fast advancement of deep learning, skeleton acquisition is not limited to use motion capture systems [25] and depth cameras [37]. The RGB data, for instance, can be used to infer 2D skeletons [1, 33] or 3D skeletons [13, 24] in real time. Moreover, even WiFi signals can be used to estimate skeleton data [30, 38]. Those achievements have made skeleton-based action recognition available on a huge amount of multimedia resources and therefore have stimulated the model's development.

In general, in order to achieve a better performance for skeleton-based action recognition, previous studies attempt to work on two aspects: introduce new features for skeleton sequences [2, 4, 5, 7, 8, 21, 36], and, propose novel neural network architectures [10, 14, 17, 18, 23, 29, 35].

A good skeleton-sequence representation should contain global motion information and be location-viewpoint invariant. However, it is challenging to satisfy both requirements in one feature. The studies [2, 5, 7, 8] focused on global motions without considering the location-viewpoint variation in their features. Other studies [4, 21, 36], on the contrary, introduced location-viewpoint invariant features without considering global motions. Our work bridges their gaps by seamlessly integrating a location-viewpoint invariant feature and a two-scale global motion feature together.

Although Recurrent Neural Networks (RNNs) are commonly used in skeleton-based action recognition [11, 19, 20, 28, 31, 36], we argue that it is relatively slow and difficult for parallel computing, compared with methods [5, 10, 18] that use Convolutional Neural Networks (CNNs). Since we take the model speed as one of our priorities, we utilize 1D CNNs to construct the backbone network of DD-Net.

3 METHODOLOGY

The network architecture of Double-feature Double-motion Network (DD-Net) is shown in Fig. 2. In the following, we explain our motivation for designing input features and network structures of DD-Net.

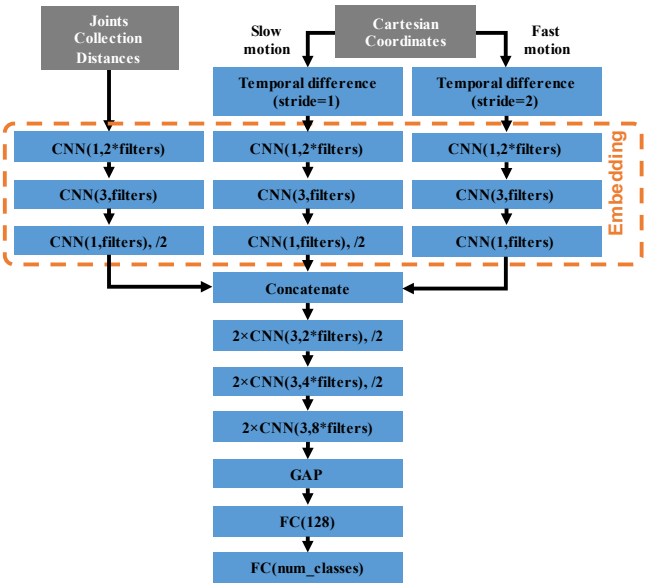

Figure 2: The network architecture of DD-Net. "2×CNN(3, 2*$filters$), /2" denotes two 1D ConvNet layers (kernel size = 3, channels = 2*$filters$) and a Maxpooling (strides = 2). Other ConvNet layers are defined in the same format. GAP denotes Global Average Pooling. FC denotes Fully Connected Layers (or Dense Layers). We can change the model size by modifying $filters$.

3.1 Modeling Location-viewpoint Invariant Feature by Joint Collection Distances (JCD)

For skeleton-based action recognition, two types of input features are commonly used: the geometric feature [4, 36] and the Cartesian coordinate feature [14, 28, 31, 34, 39]. The Cartesian coordinate feature is variant to locations and viewpoints. As Fig. 1 (a) shows, when skeletons are rotated or shifted, the Cartesian coordinate feature can be significantly changed. The geometric feature (*e.g.*, angles/distances), on the other hand, is location-viewpoint invariant, and thereby it has been utilized for skeleton-based action recognition for a while. However, existing geometric features may need to be heavily redesigned from one dataset to another [4, 36], or, contain redundant elements [19]. To alleviate these issues, we introduce a Joint Collection Distances (JCD) feature.

We calculate the Euclidean distances between a pair of collective joints to obtain a symmetric matrix. To reduce the redundancy, only the lower triangular matrix without the diagonal part is used as the JCD feature (see Fig. 3). Hence, the JCD feature is less than half the size of [19].

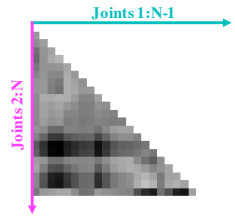

Figure 3: An example of Joint Collection Distances (JCD) feature at frame k, where the number of joints is N.

In more detail, we assume the total frame number is K ($K = 32$ as the default setting) and there are totally N joints for one subject. At frame k, the 3D Cartesian coordinates of joint n is represented as $J_i^k = (x, y, z)$, while the 2D Cartesian coordinates is represented as $J_i^k = (x, y)$. Put all of joints together, we have a joint collection $S^k = \{J_1^k, J_2^k, ..., J_N^k\}$. The formula for calculating the JCD feature of S^k is:

$$JCD^k = \begin{bmatrix} \left\|\overrightarrow{J_2^k J_1^k}\right\|_2 & & & \\ \vdots & \ddots & & \\ \vdots & \cdots & \ddots & \\ \left\|\overrightarrow{J_N^k J_1^k}\right\|_2 & \cdots & \cdots & \left\|\overrightarrow{J_N^k J_{N-1}^k}\right\|_2 \end{bmatrix}; \quad (1)$$

where $\left\|\overrightarrow{J_i^k J_j^k}\right\|_2 (i \neq j)$ denotes the Euclidean distance between J_i^k and J_j^k.

In our processing, the JCD feature is flattened to be a one-dimensional vector as our model's input. The dimension of flattened JCD is $\binom{N}{2}$.

3.2 Modeling Global Scale-invariant Motions by a Two-scale Motion Feature

Although the JCD feature is location-viewpoint invariant, the same as other geometric features, it does not contain global motion information. When actions are associated with global trajectories (see Fig. 1 (c)), solely using the JCD feature is insufficient. Unlike previous works that only utilize either the geometric feature [4, 36] or the Cartesian coordinate feature [17, 18, 29, 35], our DD-Net seamlessly integrates both of them.

We calculate the temporal differences (*i.e.*, the speed) of the Cartesian coordinate feature to obtain global motions, which is location-invariant. For the same action, however, the scale of global motions may not be exactly identical. Some might be faster, and others might be slower (see Fig. 1 (b)). To learn a robust global motion feature, both fast and slow motions should be considered. Conferring this intuition to DD-Net, we employ a fast global motion and a slow global motion to form a two-scale global motion feature. This idea is inspired by the two-scale optical flows proposed for RGB-based action recognition [12].

Technically, the two-scale motions can be generated by the following equation:

$$\begin{aligned} M_{slow}^k &= S^{k+1} - S^k \text{ for } k \in \{1, 2, 3, ..., K-1\}; \\ M_{fast}^k &= S^{k+2} - S^k \text{ for } k \in \{1, 3, ..., K-2\}; \end{aligned} \quad (2)$$

where M_{slow}^k and M_{fast}^k denote the slow motion and the fast motion at frame k, respectively. S^{k+1} and S^{k+2} are behind the S^k of one frame and two frames, respectively. Corresponding to $S^{[1,...,K]}$, we have $M_{slow}^{[1,...,K-1]}$ and $M_{fast}^{[1,...,K/2-1]}$ when K is an even number.

To generate an one-dimensional input at each frame, we reshape M_{slow}^k and M_{fast}^k as $M_{slow}^k \in \mathbb{R}^{D_{motion}}$ and $M_{fast}^k \in \mathbb{R}^{D_{motion}}$, respectively, where D_{motion} is the dimension of flattened vector. To match the frame number of the JCD feature, we perform linear interpolation to resize $M_{slow}^{[1,...,K-1]}$ and $M_{fast}^{[1,...,K/2-1]}$ as $M_{slow}^{[1,...,K]}$ and $M_{fast}^{[1,...,K/2]}$, respectively. Consequently, two-scale global motion feature is composed of $M_{slow}^{[1,...,K]} \in \mathbb{R}^{K \times D_{motion}}$ and $M_{fast}^{[1,...,K/2]} \in \mathbb{R}^{(K/2) \times D_{motion}}$. Such a process can be done in our DD-Net, and only the Cartesian coordinate feature is needed as the input.

3.3 Modeling Joint Correlations by an Embedding

Fig. 1 (d) shows that the joint indices (*i.e.*, the IDs of the head, left and right hands, *etc.*) are not locally correlated. Moreover, in different actions, the correlation of joints could be dynamically changed. Hence, the difficulty arises when we try to pre-define the correlation of joints by manually ordering their indices.

Since most of neural networks inherently assume that inputs are locally correlated, directly processing the locally uncorrelated joint feature is inappropriate. To tackle this problem, our DD-Net embeds the JCD feature and the two-scale motion feature into latent vectors at each frame. The correlation of joints is automatically learned through the embedding. As another benefit, the embedding process also reduces the effect of skeleton noise.

More formally, let embedding representations of JCD^k, M_{slow}^k and M_{fast}^k to be ε_{JCD}^k, $\varepsilon_{M_{slow}}^k$ and $\varepsilon_{M_{fast}}^k$, respectively, the embedding operation is as follows,

$$\begin{aligned} \varepsilon_{JCD}^k &= Embed_1(JCD^k); \\ \varepsilon_{M_{slow}}^k &= Embed_1(M_{slow}^k); \\ \varepsilon_{M_{fast}}^k &= Embed_2(M_{fast}^k). \end{aligned} \quad (3)$$

where the $Embed_1$ is defined as $Conv1D(1, 2*filters) \rightarrow Conv1D(3, filters) \rightarrow Conv1D(1, filters)$, and the $Embed_2$ is defined as $Conv1D(1, 2*filters) \rightarrow Conv1D(3, filters) \rightarrow Conv1D(1, filters) \rightarrow Maxpooling(2)$, because JCD^k and M_{slow}^k have twice the temporal length of M_{fast}^k.

DD-Net futher concatenates embedding features to a representation ε^k by

$$\begin{aligned} \varepsilon^k &= \varepsilon_{JCD}^k \oplus \varepsilon_{M_{slow}}^k \oplus \varepsilon_{M_{fast}}^k, \\ w.r.t.\ \varepsilon^k &\in \mathbb{R}^{(K/2) \times filters}; \end{aligned} \quad (4)$$

where $\oplus$ is the concatenation operation.

After the embedding process, subsequent processes are not affected by the joint indices, and therefore DD-Net can use the 1D ConvNet to learn the temporal information as Fig. 2 shows.

4 EXPERIMENTS

4.1 Experimental Datasets

We select two skeleton-based action recognition datasets, as SHREC dataset [9] and JHMDB dataset [15], to evaluate our DD-Net from different perspectives (see Table 1).

Table 1: Properties of experimental datasets

	SHREC Dataset	JHMDB Dataset
Number of samples	2,800	928
Training/ Testing Setup	1 Training Set 1 Testing Set	3 Split Training/ Testing Sets
Dimension of skeletons	3	2
subject	hand	body
Number of actions	14 and 28	21
Actions are strongly correlated to global trajectories	✓	✗

Although other information (*e.g.*, RGB data) is available, only the skeleton information is used in our experiments. 3D skeletons are given by SHREC dataset, which are derived from RGB-D data and contain more spatial information. In JHMDB dataset, 2D skeletons are interpreted from RGB videos, which can be applied in more general cases where inferring the depth information may be hard or impossible. Besides, actions in SHREC dataset are strongly correlated to the subject's global trajectories (*e.g.*, a hand swipes a 'V' shape), while JHMDB dataset may have a weak connection with global trajectories. We show how these properties affect the performance and demonstrate the generalization of DD-Net in our ablation studies.

4.2 Evaluation Setup

The SHREC dataset is evaluated in two cases: 14 gestures and 28 gestures. The JHMDB dataset is evaluated by using the manually annotated skeletons, and we average the results from three split training/testing sets.

In ablation studies, we explore how each DD-Net component contributes to the action recognition performance by removing one component while remaining the others unchanged. Furthermore, we also explore how the performance varies with different model sizes by adjusting the value of $filters$ in Fig. 2.

4.3 Implementation Details

Since the DD-Net is small, it is feasible to put all of the training sets into one batch on a single GTX 1080Ti GPU. We choose Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [16] as the optimizer, with an annealing learning rate that drops from 1^{-3} to 1^{-5}. During the training, DD-Net only takes a temporal augmentation, which randomly selects 90% of all frames.

To demonstrate the superiority of DD-Net, we do not apply any ensemble strategy or pre-trained weights to boost the performance. To make DD-Net easily deployable to real applications, we implement it by Keras [6] with Tensorflow backend, which is "notorious" for its slow execution speed. Using other neural network frameworks may make DD-Net even faster.

4.4 Result Analysis and Discussion

The action recognition results of SHREC dataset are presented in Table 2 and more details are listed in their confusion matrix. The confusion matrix of 14 actions and 28 actions are Fig. 4 and Fig. 5, respectively. The action recognition results of JHMDB dataset are presented in Table 3.

Table 2: Results on SHREC (Using 3D skeletons only)

Methods	Parameters	14 Gestures	28 Gestures	Speed on GPU
Dynamic hand [8] (CVPRW16)	-	88.2%	81.9%	-
Key-frame CNN [9] (3DOR17)	7.92 M	82.9%	71.9%	-
3 Cent [2] (STAG17)	-	77.9%	-	-
CNN+LSTM[26] (PR18)	8-9 M	89.8%	86.3%	238 FPS
Parallel CNN [10] (RFIAP18)	13.83 M	91.3%	84.4%	-
STA-Res-TCN [14] (Gesture18)	5-6 M	93.6%	90.7%	303 FPS
MFA-Net [5] (Sensor19)	-	91.3%	86.6%	361 FPS
DD-Net (filters=64, w/o global fast&slow motion)	1.70 M	55.2%	41.6%	-
DD-Net (filters=64, w/o global slow motion)	1.76 M	92.7%	90.2%	-
DD-Net (filters=64, w/o global fast motion)	1.76 M	93.3%	90.5%	-
DD-Net (filters=64)	1.82 M	**94.6%**	**91.9%**	2,200 FPS
DD-Net (filters=32)	0.50 M	93.5%	90.4%	3,100 FPS
DD-Net (filters=16)	**0.15 M**	91.8%	90.0%	**3,500 FPS**

Table 3: Results on JHMDB (Using 2D skeletons only)

Methods	Parameters	Manually annotated skeletons	Speed on GPU
Chained Net [39] (ICCV17)	17.50 M	56.8%	33 FPS
EHPI [23] (ITSC19)	1.22 M	65.5%	29 FPS
PoTion [7] (CVPR18)	4.87 M	67.9%	100 FPS
DD-Net (filters=32, w/o global fast&slow motion)	0.46 M	71.4%	-
DD-Net (filters=32, w/o global slow motion)	0.48 M	74.9%	-
DD-Net (filters=32, w/o global fast motion)	0.48 M	75.8%	-
DD-Net (filters=32)	0.50 M	**78.0%**	3,100 FPS
DD-Net (filters=64)	1.82 M	77.8%	2,200 FPS
DD-Net (filters=16)	**0.15 M**	74.7%	**3,500 FPS**

Overall, although DD-Net takes fewer parameters, it can achieve superior results on SHREC dataset and JHMDB dataset. The confusion matrix also shows that DD-Net is robust to each action class. Despite the data property divergence existing, DD-Net demonstrates its generalization ability, which suggests it can accommodate a wide range of skeleton-based action recognition scenarios.

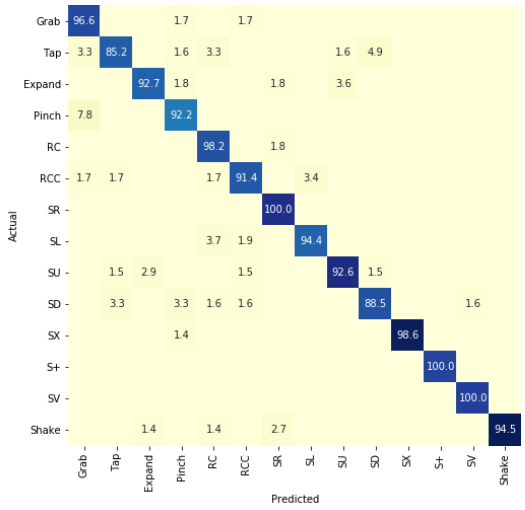

Figure 4: Confusion matrix of SHREC dataset (14 hand actions) obtained by DD-Net.

From ablation studies, we can inspect that when actions are strongly correlated to global trajectories (*e.g.*, SHREC dataset), just using the JCD feature cannot produce a satisfactory performance. When actions are not strongly correlated to global trajectories (*e.g.*, JHMDB dataset), the global motion feature still helps to improve the performance, but not as significant as the previous case. Such results agree with our assumptions: although the JCD feature is location-viewpoint invariant, it is isolated from global motions. The results also show that using the two-scale motion feature generates higher classification accuracy than only using a one-scale motion feature, which suggests that our proposed two-scale global motion feature is more robust to scale variation of motions. With the same components, DD-Net can adjust its model size by modifying the value of $filters$ in CNN layers. We select 64, 32 and 16 as the values of $filters$ to perform experiments. When DD-Net reaches the best performance on SHREC and JHMDB datasets, the values of $filters$ are 64 and 32, respectively. It is worth noting that DD-Net can generate comparable results by only using 0.15 *million* parameters.

In addition, since DD-net employs one-dimensional CNNs to extract the feature, it is much faster than other models that use RNNs [17, 28, 31, 36] or 2D/3D CNNs [7, 10, 22, 23, 39]. During its inferences, DD-Net's speed can reach around 3,500 FPS on one GPU (*i.e.*, GTX 1080Ti), or, 2,000 FPS on one CPU (*i.e.*, Intel E5-2620). While RNN-based models face great challenges for parallel processing (due to sequential dependency), our DD-Net does not have this issue because CNNs are used. Therefore, whether low-computational (*e.g.*, on small devices) or high-computational applications (*e.g.*, on parallel computing stations) are concerned, our DD-Net enjoys significant superiority.

5 CONCLUSION

By analyzing the basic properties of skeleton sequences, we propose two features and a DD-Net for efficient skeleton-based action recognition. Although DD-Net only contains a few parameters, it can achieve state-of-the-art performance on our experimental datasets. Due to the simplicity of DD-Net, many possibilities exist to enhance/extend it for broader studies. For instance, online action recognition can be approached by modifying the frame sampling strategies; RGB data or depth data could be used with it to further improve the action recognition performance; it is also possible to extend it for temporal action detection by adding temporal segmentation related modules.

6 ACKNOWLEDGEMENTS

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237, the Royal Society under IEC\R3\170013 - International Exchanges 2017 Cost Share (Japan and Taiwan only), and a MSRA Collaborative Research 2019 Grant by Microsoft Research Asia.

REFERENCES

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 1302–1310.
[2] Fabio Marco Caputo, Pietro Prebianca, Alessandro Carcangiu, Lucio D Spano, and Andrea Giachetti. 2017. A 3 Cent Recognizer: Simple and Effective Retrieval and Classification of Mid-air Gestures from Single 3D Traces. *Smart Tools and Apps for Graphics. Eurographics Association* (2017).
[3] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. 2011. A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities* 32, 6 (2011), 2566–2570.
[4] Cheng Chen, Yueting Zhuang, Feiping Nie, Yi Yang, Fei Wu, and Jun Xiao. 2011. Learning a 3D human pose distance metric from geometric pose descriptor. *IEEE Transactions on Visualization and Computer Graphics* 17, 11 (2011), 1676–1689.
[5] Xinghao Chen, Guijin Wang, Hengkai Guo, Cairong Zhang, Hang Wang, and Li Zhang. 2019. MFA-Net: Motion Feature Augmented Network for Dynamic Hand Gesture Recognition from Skeletal Data. *Sensors* 19, 2 (2019), 239.
[6] François Chollet et al. 2015. Keras.
[7] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. 2018. PoTion: Pose MoTion Representation for Action Recognition. In *CVPR 2018*.
[8] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. 2016. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1–9.
[9] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. 2017. SHREC'17 Track: 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset. In *10th Eurographics Workshop on 3D Object Retrieval*.
[10] Guillaume Devineau, Wang Xi, Fabien Moutarde, and Jie Yang. 2018. Convolutional Neural Networks for Multivariate Time Series Classification using both Inter-and Intra-Channel Parallel Convolutions. In *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP'2018)*.
[11] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1110–1118.
[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2018. SlowFast Networks for Video Recognition. *arXiv preprint arXiv:1812.03982* (2018).
[13] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2018. Real-time 3D Hand Pose Estimation with 3D Convolutional Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).
[14] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. 2018. Spatial-Temporal Attention Res-TCN for Skeleton-based Dynamic Hand Gesture Recognition. *gesture* 30, 5 (2018), 3.
[15] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. 2013. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*. 3192–3199.
[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. 2017. Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1012–1020.
[18] Chuankun Li, Yonghong Hou, Pichao Wang, and Wanqing Li. 2017. Joint Distance Maps Based Action Recognition with Convolutional Neural Network. *IEEE Signal Processing Letters* 24, 5 (2017), 624–628.

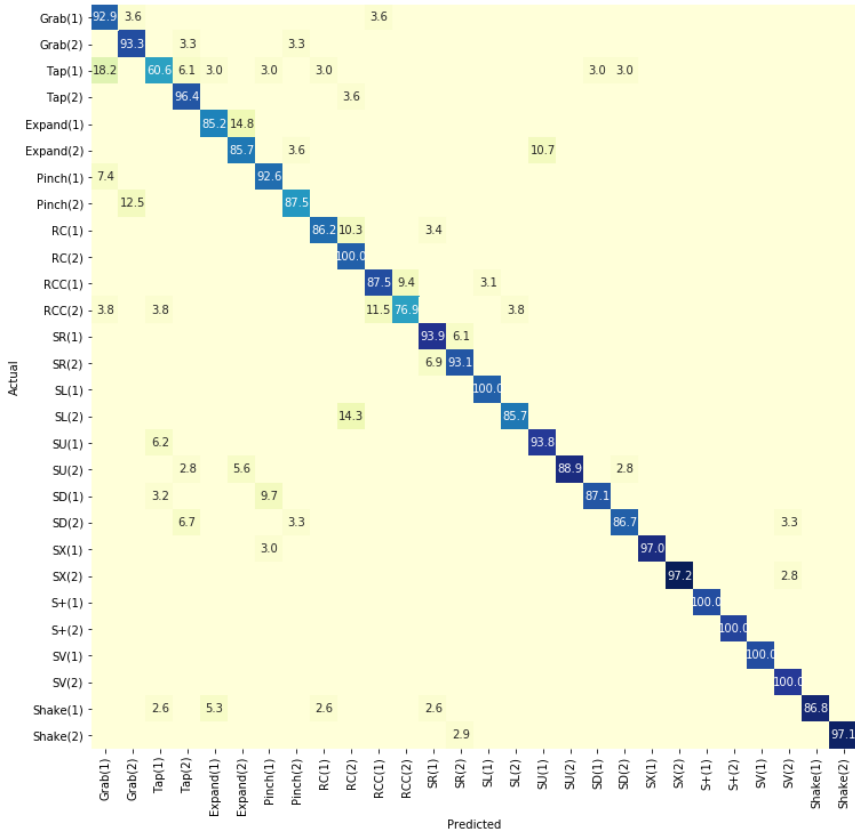

Figure 5: Confusion matrix of SHREC dataset (28 hand actions) obtained by DD-Net.

[19] Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. 2017. Skeleton-based action recognition using LSTM and CNN. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*. IEEE, 585–590.

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

[21] Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* 68 (2017), 346–362.

[22] Mengyuan Liu and Junsong Yuan. 2018. Recognizing Human Actions as the Evolution of Pose Estimation Maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1159–1168.

[23] Dennis Ludl, Thomas Gulde, and Crist'obal Curio. 2019. Simple yet efficient real-time pose-based action recognition. In *ITSC*.

[24] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics* 36, 4, 14. http://gvv.mpi-inf.mpg.de/projects/VNect/

[25] Thomas B Moeslund and Erik Granum. 2001. A survey of computer vision-based human motion capture. *Computer vision and image understanding* 81, 3 (2001), 231–268.

[26] Juan C Nunez, Raul Cabido, Juan J Pantrigo, Antonio S Montemayor, and Jose F Velez. 2018. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition* 76 (2018), 80–94.

[27] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. 2011. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 759–760.

[28] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data.. In *AAAI*, Vol. 1. 4263–4270.

[29] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. 2018. Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5323–5332.

[30] Fei Wang, Stanislav Panev, Ziyi Dai, Jinsong Han, and Dong Huang. 2019. Can WiFi Estimate Person Pose? *arXiv preprint arXiv:1904.00277* (2019).

[31] Hongsong Wang and Liang Wang. 2017. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *e Conference on Computer Vision and Pa ern Recognition (CVPR)*.

[32] Shih-En Wei, Nick C Tang, Yen-Yu Lin, Ming-Fang Weng, and Hong-Yuan Mark Liao. 2014. Skeleton-augmented human action understanding by learning with progressively refined data. In *Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia*. ACM, 7–10.

[33] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 466–481.

[34] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455* (2018).

[35] Zhengyuan Yang, Yuncheng Li, Jianchao Yang, and Jiebo Luo. 2018. Action Recognition with Spatio-Temporal Visual Attention on Skeleton Image Sequences. *IEEE Transactions on Circuits and Systems for Video Technology* (2018).

[36] Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang. 2018. Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks. *IEEE Transactions on Multimedia* 20, 9 (2018), 2330–2343.

[37] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (2012), 4–10.

[38] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7356–7365.

[39] Mohammadreza Zolfaghari, Gabriel L Oliveira, Nima Sedaghat, and Thomas Brox. 2017. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Computer Vision (ICCV), 2017 IEEE International Conference on Computer Vision*. IEEE, 2923–2932.