Deep Learning-based Feature Fusion for Action Recognition Using Skeleton Information

Fahad Ul Hassan Asif Mattoo*, Umar Shahbaz Khan, Tahir Nawaz and Nasir Rashid
Department of Mechatronics Engineering, College of Electrical and Mechanical Engineering,
National University of Sciences and Technology, Islamabad, Pakistan
*email: hasif.mts19ceme@mts.ceme.edu.pk

2023 International Conference on Robotics and Automation in Industry (ICRAI) | 978-1-6654-6472-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRAI57502.2023.10089577

***Abstract*—Various action recognition systems have been proposed, but most of them are not feasible to be used in real-time applications. Skeleton-based action recognition has a low computational cost and is not affected by background changes. As the pose estimation models are becoming faster (almost real-time), a model was created with only 1.8M parameters named DD-net, which uses the skeleton information to predict the action. Recently an improved version of the model came out and was named TD-net. The model is very rich with geometric-based features but lacks motion-based features. To overcome this we added two motion features in the model named acceleration and velocity. These features were created using second order Taylor's approximation, in a window around the current frame. The model accuracy was compared with DD-net, TD-net, and state-of-the-art algorithms using three different datasets. An increase in accuracy is observed for all three datasets (i.e 1.1% for SHERC, 1.7% for FPHAB and 2% for JHMDB) when compared with TD-net.**

***Keywords*—Action recognition; Sequential skeleton information; Deep neural network; Motion features; Geometric features; Feature fusion**

I. INTRODUCTION

Action recognition systems use data from various types of sensors to analyze and identify activities. An action can be defined as a movement of a person's body parts (1 or many). This can be a single motion or a sequence of actions being performed in a predefined order. Vision-Based action recognition has become a very hot topic in the computer vision community. Important applications are being developed, including automatic video surveillance for security [1], remote healthcare monitoring [2], and virtual reality [3]. Activities can be self-contained or involve human-to-human or human-to-object interaction. There are six subcategories of activities: action, gesture, behavior, interaction (both human to human and object), and group activity [4].

VAR uses computer vision techniques to identify activities in a video sequence or image. VAR has the advantage over wearable sensors-based AR as we are not bounded by sensors mounted on the body. The research of Johansson [5] shows that vision-based systems can detect the direction of the motion along with various motion patterns for limbs. Johansson's work inspired most of the literature for human pose estimation and action recognition [6]. Skeleton-based approaches when compared with RGB-based approaches, have advantages as they have low computational and memory requirements, plus skeleton data is very discriminative when representing action. There is no change in skeleton data due to background changes. With the libraries like Open pose [7], Alpha Pose [8], Hyper Pose [9], Blaze Pose [10], and Yolov7 [11] almost real-time skeleton data can be obtained from a video stream. Due to all these advantages of skeleton data, a lot of attention has been toward approaches that use skeleton-based action recognition.

Many approaches are present in the literature for AR using skeleton key points. The two main categories are handcrafted-based and learning-based features engineering. For the first category, features are extracted manually. In [12] features were extracted using histograms of 3D joint positions. A new feature was suggested by [13] that merged static pose, motion, and offset of action. A descriptor using covariance was proposed by [14] that captured the variation of joints over time. CovMIJ was proposed by [15] that utilized Most Informative Joints (MIJ), which represented features using covariance descriptor and noise immunity. Relative and temporal derivatives of joint positions were used by [16]. Joint velocity and position covariance descriptors were combined by [17] which were extracted from MIJ.

It is not possible to make a feature descriptor for all datasets using handcrafted approaches. Deep learning can automate the process of feature extraction, representation, and classification using data in the raw state. [18] proposed a synthesized CNN, that has three stages, in the first stage a view-invariant transform is made using sequences, then a series of images are created using the modified skeletons, and a CNN-based model extract characteristics that are classified at the decision level after deep fusion of features. A motion CNN was proposed by [19] that used skeleton joints from two consecutive frames along with the joint positions. A bi-directional RNN was proposed by [20], the skeleton is divided into 5 subparts representing different parts of the physical appearance, and fed to individual subnets. Then the subnet representation is merged and fed as input to the higher layers for the final decision. Long Short-Term Memory (LSTM) was used by [21]. A graph convolution network (GCN) was proposed by [22] that used adaptive and attention-enhanced functions. However, CNN and RNN mainly use image grids or vector sequences respectively, hence they cannot exploit skeleton data structure in depth. Were GCN's based approach is computationally complex. Most of these approaches suffer from high execution time or very large model sizes hence they are not feasible in real-time

TABLE I. SHAPE OF LAYERS AT VARIOUS STAGES

Stage	JCD	NCJ	Slow Motion	Fast Motion	Velocity	Acceleration
Input Feature	32 x J	32 X K x D	32 X K x D	16 X K x D	32 X K x D	32 X K x D
Conv1D (1, 2*filters)	32 X 128	32 X 128	32 X 128	16 X 128	32 X 128	32 X 128
Conv1D (3, filters)	32 X 64	32 X 64	32 X 64	16 X 64	32 X 64	32 X 64
Conv1D (1, filters)				16 X 64		
Conv1D (1, filters)/2	16 X 64	16 X 64	16 X 64		16 X 64	16 X 64
Concatenate	16 X 384					
2 x Conv1D (3, 2*filters)/2	8 X 128					
2 x Conv1D (3, 4*filters)/2	4 X 256					
2 x Conv1D (3, 8*filters)	4 X 512					
Global average pooling	512					
FC (128)	128					
FC (Num-classes)	Classes					

applications. A lightweight model proposed by [23], can work in real-world applications as it has only 1.8M parameters, and works at a staggering 2,200 Frames per second (FPS). The authors have implemented a simple model with three types of features that are embedded in the model, these are joint collection distances and slow and fast features. This model was improved by [24] by adding a normalized cartesian joint as a feature.

The proposed work is focused on further adding new features to this model. After experimenting with a different set of features we have seen an improvement by adding acceleration and velocity as features to the existing model. Experiments were done using datasets SHREC [25], FPHAB [26], and JHMDB [27].

The paper is organized as: Section 2 explains the process by which different features are made, section 3 presents the implementation steps for training and datasets, then in section 4 experimentation results are presented, and in the final section, we give our conclusion and the future direction of the research.

II. METHODOLOGY

In the network DD-net (Double-feature Double-motion Network) [23], authors have used one geometric feature named Joint Collection Distances (JCD), and two motion features called Slow Motion, and Fast Motion features, in [24] authors added another geometric feature named Normalized Cartesian Joints (NCJ) and named there model TD-net (Triple-feature Double-motion Network). In the proposed work, we have added two new motion features, that are speed and acceleration. These features are inspired by the work of [28]. Notations in the paper are as follows:

- Number of skeletons in a sequence is represented as $\boldsymbol{S}$;
- Number of joints in a skeleton is represented as $\boldsymbol{K}$;
- Depth of skeleton is represented as $\boldsymbol{D}$;
- Total number of JCD features is represented as $\boldsymbol{J}$, and calculated as $\boldsymbol{J} = \frac{K \times (K-1)}{2}$;
- 3D coordinates are represented as $B_i^s = (x_i^s, y_i^s, z_i^s)$ for the i^{th} joint in the s^t frame;
- Collection of K joints is represented as $C^s = (B_1^s, B_2^s, ..., B_K^s)$ for the s^t frame.

A. JCD Features

Joint Collection Distance (JCD) is a geometric type of feature that is invariant to location viewpoint and was proposed by [23]. A matrix is calculated for this feature. The values in the matrix are the euclidean distance between pairs of joints. We only consider the values below the diagonal as, values on the diagonal are zero, and values above the diagonal are duplicates. This is done to remove redundant values. As shown in Fig. 1, boxes in blue are considered JCD features. This matrix is computed for each frame, and each matrix is flattened to a 1D vector which is the input to the model. Equation 1 is the numerical representation of the JCD feature for s^{th} frame.

$$JCD^s = \begin{bmatrix} \|\overrightarrow{B_2^s B_1^s}\| & & & \\ \vdots & \ddots & & \\ \vdots & \dots & \ddots & \\ \|\overrightarrow{B_K^s B_1^s}\| & \dots & \dots & \|\overrightarrow{B_K^s B_{K-1}^s}\| \end{bmatrix} \quad (1)$$

Here $\|\overrightarrow{B_i^s B_j^s}\|$ is the Euclidean distance between B_i^s and B_j^s while $(i \neq j)$.

B. Motion Feature (Slow and Fast)

To monitor changes in the temporal domain, motion-relevant features are extracted from cartesian coordinates. Features are

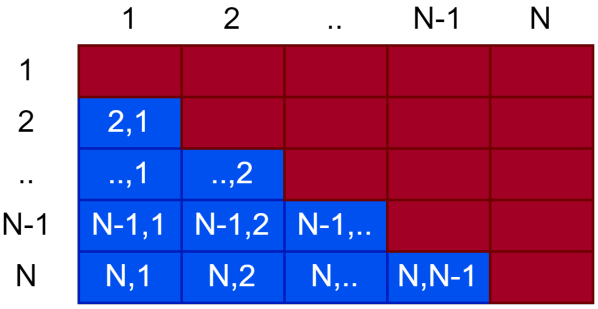


Fig. 1. JCD features matrix representation

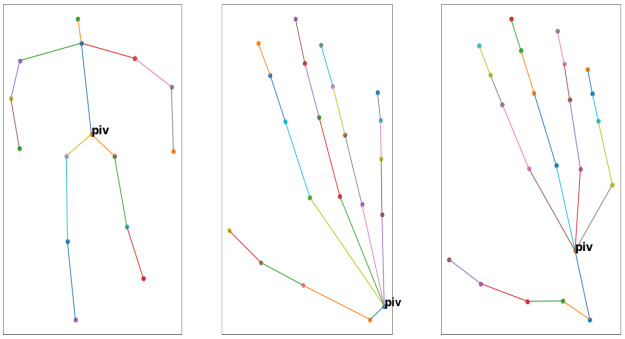
extracted in terms of slow and fast motion and numerically they are computed using equations 2 and 3 respectively.

$$M^k_{slow} = C^{s+1} - C^s, s \in 1, 2, 3, ..., S-1; \quad (2)$$

$$M^k_{fast} = C^{s+2} - C^s, s \in 1, 2, 3, ..., S-2; \quad (3)$$

Here M^k_{slow} and M^k_{fast} denote the two motion features. The output from the equations is reshaped to a 1D vector. Then linear interpolation is applied to resize M^k_{fast} as $M^{[1,...,K/2]}_{fast}$ and M^k_{slow} as $M^{[1,...,K]}_{fast}$. This is done to match the input feature of slow and fast motion with other features.

C. Normalized Cartesian Joints Features



(a) JHMDB (b) FPHAB (c) SHREC

Fig. 2. Skeleton with the marked pivot point for the three datasets

To increase spatial information for actions with weak global trajectories, NCJ features are utilized. We normalize the cartesian coordinates with respect to a pivot point. In the case of a human skeleton (JHMDB), the pivot point is hip joint. For hand skeleton, pivot point is wrist joint for FPHAB and palm joint for SHREC. The pivot point for the three datasets is show in Fig. 2. Equation 4 is used to extract NCJ for each joint.

$$NCJ^s_i = B^s_i - B^s_{piv} \quad (4)$$

The obtained sequence is flattened which is fed to the model as input.

D. Velocity and Acceleration Features

Motion features are just using two consecutive frames (slow), or one frame skipped (fast), which does not give an in-depth information for the joints kinematics. To capture this information we proposed two features that are velocity and acceleration. Velocity is used to define the speed and direction of the joints and change in speed is captured by acceleration.

Skeleton joints sequences can be considered as continuous and differentiable over time. The second-order Tayler approximation can be considered as a window around the current time step (s). The partial derivatives ∂C^s and $\partial^2 C^s$ can be defined by expanding around the current pose C^s. Equation 5 and 6 are used to estimate the partial derivatives that are called velocity and acceleration respectively. We are using a temporal window of 5 frames, that is centered at the current frame C^s being processed.

$$\partial C^s = V^s = C^{c+1} - C^{s-1}, s \in 2, 3, 4, ..., S-1; \quad (5)$$

$$\partial^2 C^s = A^s = C^{s+2} + C^{s-2} - 2C^s, s \in 3, 4, 5, ..., S-2; \quad (6)$$

Where V^k and A^k denotes the velocity and acceleration. The results are reshaped to a 1D vector and interpolation is applied similar to slow and fast motion features to keep the dimension of the features consistent with the other features.

E. Joint Correlation Feature

To correlate dynamically changing joints for different actions and datasets, embedding bock are used. There are embedding blocks for each input vector. The embedding blocks uses one Dimensional convolution layer (1D convNet) with a different number of filters to learn the correlation of the input features for each input vector. All the outputs from individual embedding blocks are concatenated at the final stage of the embedding module. As shown in Fig. 3. Detailed parameters of the model are given in Table I.

III. EXPERIMENTATION

Three different datasets are used to conduct the experiments. The data sets are split into two sets, training, and testing. Each dataset is evaluated individually, based on the calculated accuracy with respect to its classes. Hyperparameters are tuned to obtain the best possible results.

A. Datasets Used

We selected three skeleton-based action recognition datasets, SHREC [25], JHMDB [27] and FPHAB [26]. Various properties of datasets are given in Table II. RGB data for these datasets are available but we have only used the skeleton information for our experiments. Each sample represents sequential skeleton information. This means if we have a video of 10 frames, then that sample will have 10 skeletons. The SHERC and FPHAB skeleton is in 3D, and have 22 and 21 joints respectively. While the JHMDB skeleton is in 2D and has 15 joints.

B. Train/Test Splits

The train/test splits are made according to the distribution specified by the authors of the three datasets. The sample numbers for each dataset for the two splits are mentioned in Table III and their split ratio is as follows.

- SHERC a 70/30 split is made.
- FPHAB a 50/50 split is made.
- JHMDB a 70/30 split is made, with three different combinations of samples.

C. Setup for Evaluation

There are two cases for evaluating SHERC dataset. First is the 14 gesture case and second is the 28 gestures case. JHMDB is evaluated with 3 different combinations of data distributions and results are the averaged. The FPHAB dataset is evaluated for the 45 classes using the normalized data provided.

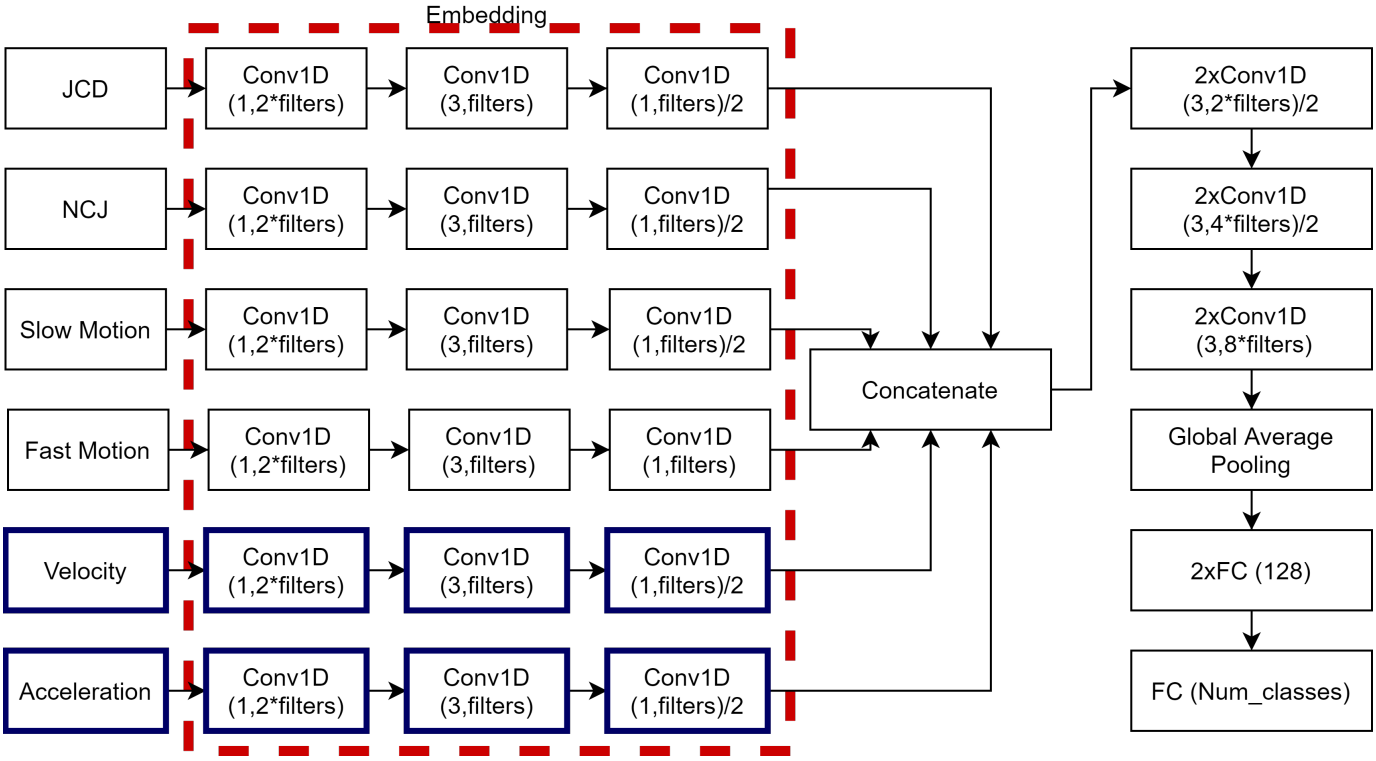


Fig. 3. Proposed architecture. New branches are marked in Blue. Fc (fully connected layers). "2xConv1D (3,2*filters)/2" (two 1D CNN layers with kernel size=3 & channel=2*filter with a Maxpooling of stride 2. Filter is set as 64.

TABLE II. PROPERTIES OF DATASETS

Property/ Datasets	Number of Samples	Joints Count	Skeleton Depth (D)	Type of Dataset	Action Count
SHERC	2,800	22	3	Hand	14 & 28
JHMDB	928	15	2	Body	21
FPHAB	1,175	21	3	Hand	45

TABLE III. SAMPLES DISTRIBUTION

Dataset	Train	Test
SHERC	1960	840
FPHAB	600	575
JHMDB	660	268
	658	270
	663	265

D. Implementation Details

Since the network is small we are training with a batch size equal to the number of samples. Google Colaboratory is being used for training/testing and the GPU being utilized is a Tesla P100. Adam optimizer is used with the parameters $\beta 1$ = 0.9, and $\beta 2$ = 0.999. The learning rate is set to $1e^{-3}$ for the first 600 epochs and $1e^{-4}$ for the last 600 epochs.

IV. RESULTS

A. Accuracy Comparison

From Table IV an improvement of 0.7% and 1.1% for the 14 and 28 actions respectively, of SHERC dataset, For JHMDB dataset accuracy is improved by 2% and for FPHAB 1.7% improvement in accuracy is seen when compared with TD-net accuracies. Comparison with state of the art algorithms is given in Table V, VI and VII for SHERC, JHMDB and FPHAB datasets respectively.

B. Parameter Comparison

In Table VIII we can see that there is an average increase of 6% in parameters when compared with TD-net and increase of 9% when compared with DD-net. Originally the DD-net is about 1.82M parameters and TD-net is 1.88M parameters.

C. Discussion

Two new features were added to the model which increased the parameters slightly, from 1.8 million to 2 million. The additional parameters do not have any effect on the execution time. Secondly, a reliable annotation of the pose points is required, when some pose points are missing method might fail for some classes. From Table VIII we can see the total parameters for our model are around 2 million, from which we can say this is a relatively small model as compared to models

TABLE IV. ACCURACY COMPARISON WITH DD-NET & TD-NET

Method	Accuracy			
	SHERC		JHMDB	FPHAB
DD-Net [23]	94.6%	91.9%	77.2%	90.0%
TD-Net [24]	95.0%	92.4%	79.3%	93.2%
Proposed Model	**95.7%**	**93.5%**	**81.3%**	**94.9%**

TABLE V. SHREC RESULTS AND COMPARISON

Method	Accuracy	
	14 Gestures	28 Gestures
Key-frame CNN [25]	82.9%	71.9%
CNN+LSTM [29]	89.8%	86.3%
MFA-Net [30]	91.3%	86.6%
STA-Res-TCN [31]	93.6%	90.7%
DD-Net [23]	94.6%	91.9%
TD-Net [24]	95.0%	92.4%
Proposed Model	**95.7%**	**93.5%**

TABLE VI. JHMDB RESULTS AND COMPARISON

Method	Accuracy
STAR-net [32]	64.3%
PoTion [33]	67.9%
DD-Net [23]	77.2%
TD-Net [24]	79.3%
Proposed Model	**81.3%**

TABLE VII. FPHAB RESULTS AND COMPARISON

Method	Accuracy
LSTM 3D-GT [26]	56.8%
PA-ResGCN [34]	65.5%
DD-Net [23]	90.0%
TD-Net [24]	93.2%
Proposed Model	**94.9%**

TABLE VIII. PARAMETERS COMPARISON

Method	Datasets		
	SHERC	FPHAB	JHMDB
Proposed Model (Parameters)	2.01M		1.97M
Percentage Change from TD-Net [24]	+6.20%		+5.88%
Percentage Change from DD-Net [23]	+9.31%		+8.82%

used for image-based approaches which have parameters more than 20 million. Fig. 4 gives the general system overview of the proposed methodology.

V. CONCLUSION

The proposed study introduces a model with triple feature and quad motion network (TQ-net), as we are using three geometric features and four motion features. An improvement in accuracy is seen for all three datasets, 0.7% and 1.1% for SHERC, 1.7% for FPHAB and 2% for JHMDB. The model parameters are increased by 6-9% by the addition of the 2

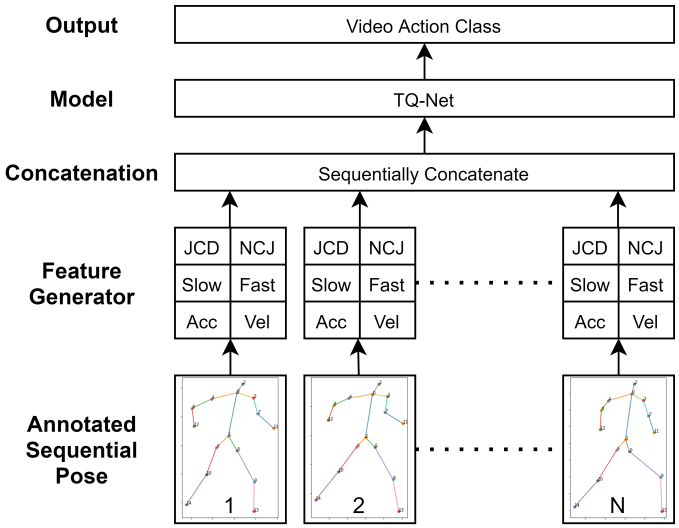


Fig. 4. System Overview

new motion features (Acceleration & Velocity). As the model is very simple with very few parameters real-time application is a possibility while utilizing pose estimation tools. In the future, we are planning on adding additional features to the model, and building an application using a live video feed and classifying using the trained model weights.

REFERENCES

[1] T. Nawaz and J. Ferryman, "An annotation-free method for evaluating privacy protection techniques in videos," in *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2015, pp. 1–6.

[2] T. Nawaz, B. Rinner, and J. Ferryman, "User-centric, embedded vision-based human monitoring: A concept and a healthcare use case," in *Proceedings of the 10th International Conference on Distributed Smart Camera*, 2016, pp. 25–30.

[3] M. J. Schuemie, P. Van Der Straaten, M. Krijn, and C. A. Van Der Mast, "Research on presence in virtual reality: A survey," *CyberPsychology & Behavior*, vol. 4, no. 2, pp. 183–201, 2001.

[4] A. B. Sargano, P. Angelov, and Z. Habib, "A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition," *applied sciences*, vol. 7, no. 1, p. 110, 2017.

[5] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[6] G. V. Kale and V. H. Patil, "A study of vision based human motion recognition and analysis," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 7, no. 2, pp. 75–92, 2016.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[8] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2334–2343.

[9] Y. Guo, J. Liu, G. Li, L. Mai, and H. Dong, "Fast and flexible human pose estimation with hyperpose," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3763–3766.

[10] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking. arxiv," *arXiv preprint arXiv:2006.10204*, 2020.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[12] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 20–27.

[13] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 14–19.

[14] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Twenty-third international joint conference on artificial intelligence*, 2013.

[15] T.-N. Nguyen, D.-T. Pham, T.-L. Le, H. Vu, and T.-H. Tran, "Novel skeleton-based action recognition using covariance descriptors on most informative joints," in *2018 10th international conference on knowledge and systems engineering (KSE)*. IEEE, 2018, pp. 50–55.

[16] D.-T. Pham, T.-N. Nguyen, T.-L. Le, and H. Vu, "Spatio-temporal representation for skeleton-based human action recognition," in *2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE, 2020, pp. 1–6.

[17] V.-T. Nguyen, T.-N. Nguyen, T.-L. Le, D.-T. Pham, and H. Vu, "Adaptive most joint selection and covariance descriptions for a robust skeleton-based human action recognition," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 27 757–27 783, 2021.

[18] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[19] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 597–600.

[20] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[21] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[22] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

[23] F. Yang, S. Sakti, Y. Wu, and S. Nakamura, "Make skeleton-based action recognition model smaller, faster and better," *CoRR*, vol. abs/1907.09658, 2019.

[24] T.-T. Nguyen, D.-T. Pham, H. Vu, and T.-L. Le, "A robust and efficient method for skeleton-based human action recognition and its application for cross-dataset evaluation," *IET Computer Vision*, 2022. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi2.12119

[25] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset," in *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 1–6.

[26] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[27] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3192–3199.

[28] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2752–2759.

[29] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018.

[30] X. Chen, G. Wang, H. Guo, C. Zhang, H. Wang, and L. Zhang, "Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data," *Sensors*, vol. 19, no. 2, p. 239, 2019.

[31] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, and H. Yang, "Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[32] W. McNally, A. Wong, and J. McPhee, "Star-net: Action recognition using spatio-temporal activation reprojection," in *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 2019, pp. 49–56.

[33] B. Ren, M. Liu, R. Ding, and H. Liu, "A survey on 3d skeleton-based action recognition using learning method," *arXiv preprint arXiv:2002.05907*, 2020.

[34] V.-D. Le, V.-N. Hoang, T.-T. Nguyen, V.-H. Le, T.-H. Tran, H. Vu, and T.-L. Le, "A unified deep framework for hand pose estimation and dynamic hand action recognition from first-person rgb videos," in *2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE, 2021, pp. 1–6.