2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-5348-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IMCOM56909.2023.10035660

# Two-Branch Stacked Transformer for 2D Skeleton-based Action Recognition

Yerassyl Zhalgasbayev
*School of Sciences and Humanities*
*Nazarbayev University*
Nur-Sultan, Kazakhstan
yerassyl.zhalgasbayev@nu.edu.kz

Nguyen Anh Tu
*School of Engineering and Digital Sciences*
*Nazarbayev University*
Nur-Sultan, Kazakhstan
tu.nguyen@nu.edu.kz

***Abstract*—Human Action Recognition (HAR) is a challenging computer vision task with various applications, ranging from smart surveillance to human-computer interaction. Recently, the human skeleton, a compact and intuitive data modality, has attracted increasing attention in many studies and has achieved good results in HAR. However, some challenges such as body occlusion and action similarity still need to be addressed. In this paper, to overcome these challenges, we propose a model combining short action-snippets for storing meaningful information about human body transition and a deep network configured by two parallel branches of Transformer for thoroughly learning the temporal correlation of skeletal representations in upper and lower body parts, hence concurrently enabling to handle of partial occlusions of skeleton data and boosting the HAR performance. In experiments, we validate the proposed approach's outperformance compared with the state-of-the-art methods on the JHMDB dataset in terms of both the size (i.e., number of learned parameters) and the accuracy.**

***Index Terms*—action recognition, 2D skeleton, action-snippet, Transformer.**

## I. Introduction

Researchers consider computer vision an essential field in Artificial Intelligence since it significantly impacts a wide variety of real-world applications. One of the most challenging tasks of computer vision is video-based human action recognition (HAR), involving the identification of different actions from videos [1]. This task is vital in many areas, such as healthcare systems, video surveillance, and crime detection. Various difficulties exist when recognizing human actions from video sequences, such as occlusion, illumination, and scarcity of labeled action data [3]. Many scientists tried different approaches to tackle these problems, but their results still need further investigation. Specifically, a large number of former works using neural networks have explored the feature representation based on RGB frames and optical flows. However, these data modalities are sensitive to background variation. In other aspects, with advancements in depth imaging technology and pose estimation algorithms, recent works have paid great attention to the human skeleton since this data modality is compact, concise, and intuitive for discriminating various actions. These attractive properties are well-suited for generating the discriminative video representation with a small memory footprint, enabling real-time processing on resource-constrained devices.

Our aim in this paper is to propose a method that yields comparable performance in accordance with state-of-the-art HAR frameworks. This method is based on 2D skeletons and short action-snippets [6]. These action-snippets are the concatenation of 2D skeletons of consecutive video frames and have shown their robustness in recognition of actions with similar frames in previous works [4]–[6]. Accordingly, we employ these action-snippets to extract more discriminative features, such as velocities and positions of joints of consecutive skeletons. Despite their great advantages, skeleton-based HAR methods are still error-prone to body occlusions. Then, inspired by [7], we introduce a two-branch deep sequential network to process upper and lower body parts separately. In this way, partial body occlusions can be handled more effectively because, in many cases, some actions (e.g., hand-waving and boxing) are recognizable by using only partial body parts (e.g., hands). The approach in [7] designed two LSTM networks in parallel to generate high-level abstractions for two body parts. However, LSTMs are incapable of fully exploiting long action sequences. Also, training LSTM is expensive and complicated. Therefore, we develop a two-branch stacked Transformer to overcome these limitations as an alternative to the LSTM-based framework. The powerful Transformer [2] has revolutionized computer vision in the past few years. Due to the use of a self-attention mechanism and positional embeddings, the Transformer network can not only capture well the temporal relationship of action-snippets but also perform training efficiently. Two branches are connected via the fusion layer to combine skeletal features in the upper and lower parts. Extensive benchmark results verify the superiority of our proposed model against other methods.

The rest of the paper contains the following information: Section II discusses works related to our approach. In Section III written detailed information about the architecture of our model. Section IV reports the results of the proposed method compared to other frameworks. Finally, Section V concludes the paper and gives intuition for further investigation.

## II. Related works

HAR approaches are divided into two, non-skeleton and skeleton based. Non-skeleton-based approaches use RGB data to learn, but their possibilities are limited due to shadows and background actions. Meantime, skeleton-based models use 2D or 3D data to make predictions. These data are advantageous due to their intuitiveness, compactness, and conciseness, as mentioned above. In some of the recent works [13], [14], authors used 3D data received from depth cameras. However, these cameras can not work properly outdoors. Therefore, most scientists started to use 2D skeletons.

In [9], [12], authors used convolutional neural networks (CNNs) to process 2D data. Whilst, in [18] was used heat-map of body joints combined with CNN. Even so, these CNN-based algorithms achieved good results, and they are incapable of capturing sequential characteristics of consecutive video frames. Consequently, in some recent works RNN and LSTM were used to 2D skeleton and achieved outstanding results [7], [15]–[17]. However, these models used simple frame-wise representations of videos and could not differentiate between complex actions. In [6], authors tried to cope with this problem by using short action-snippets and transformers, but their work still needs improvements in terms of performance. Our proposed method solved this problem by dividing body joints into two parts, which is more robust to partial body occlusions and can be beneficial in scenarios where only upper or lower body parts are enough to classify an action.

## III. Methodology

### A. Architecture overview

The general workflow of our model is shown in Figure 2 and is as follows: Firstly, the system receives skeletons extracted from video frames. These skeletons are divided into upper and lower parts, as shown in Figure 1. First $W$ skeletons in each body part are stacked together to form a list of the first action-snippet of size $W$. This list slides over all video frames by dropping the first skeleton and adding the next skeleton that is not in the list to form other snippets. Skeletons in these action-snippets are then used to compute joint-based features extracted from the same action-snippets, namely velocities and positions of joints. Then, PCA is used to reduce the dimension of feature vectors, and these feature vectors are fed into a two-branch Transformer, where each branch of the Transformer receives only upper or lower body part features. After, the outputs of these branches are concatenated using the average fusion method and processed through global max pooling and fully connected layers to receive classification probabilities.

### B. Preprocessing and Data augmentation

In our experiment, we used a dataset with already extracted skeletons. Each skeleton has 15 joints with positions in 2D space. In case of a missed joint position due to occlusion, the previous position of the same joint with respect to the neck is used to find the relative value of the missed joint position. Moreover, to tackle insufficient data, we applied different data augmentation methods as follows:

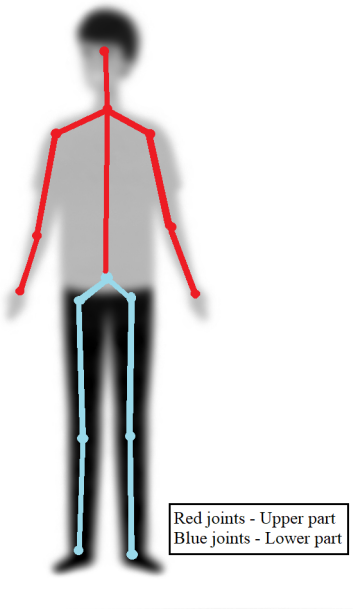


Fig. 1. Division of skeleton joints into upper and lower parts.

- Flip original data along x-axis
- Add Gaussian noise to original data with sigma $0.02$
- Add Gaussian noise to original data with sigma $0.04$
- Rotate flipped data to $15°$
- Rotate flipped data to $-15°$
- Rotate flipped data to $30°$
- Rotate flipped data to $-30°$

### C. Joint-based features extraction

Feature extraction methods are mainly based on ideas proposed in [6]. A list of size $W$ is used to combine consecutive $W$ frames' skeleton, divided into upper and lower parts, and extract joint-based features for each part. Notice, given the number of skeletons $W$ in one action-snippet, the full set of skeletal joints is defined as:

$$S_{up} = \{\xi_i^t = (x_i^t, y_i^t) \mid i \in \mathbb{J}_{up}, t \in [1, W]\} \quad (1)$$

$$S_{low} = \{\xi_i^t = (x_i^t, y_i^t) \mid i \in \mathbb{J}_{low}, t \in [1, W]\} \quad (2)$$

where $\mathbb{J}_{up}$ and $\mathbb{J}_{low}$ are the set of upper and lower body joints divided as in Figure 1. If $N$ is a total number of joints, then $N = |\mathbb{J}_{up}| + |\mathbb{J}_{low}|$. Moreover, note that:

- $\xi_i^t$ is a position of $i$-th joint at frame $t$
- $\xi_0^t$ is a position of neck at frame $t$

Hereafter, any subscripts such as $up$ and $low$ are not be used and $S$ or $\delta$ means $S_{up}$ or $\delta_{up}$, respectively. Since, the feature extraction methods described below are the same for the both parts of the body joints.

First, to extract joint-based features we calculated average height ($\hat{H}$) of skeletons in an action-snippet, which is needed to normalize joint positions. Height of skeleton is defined as a distance between neck and thigh.

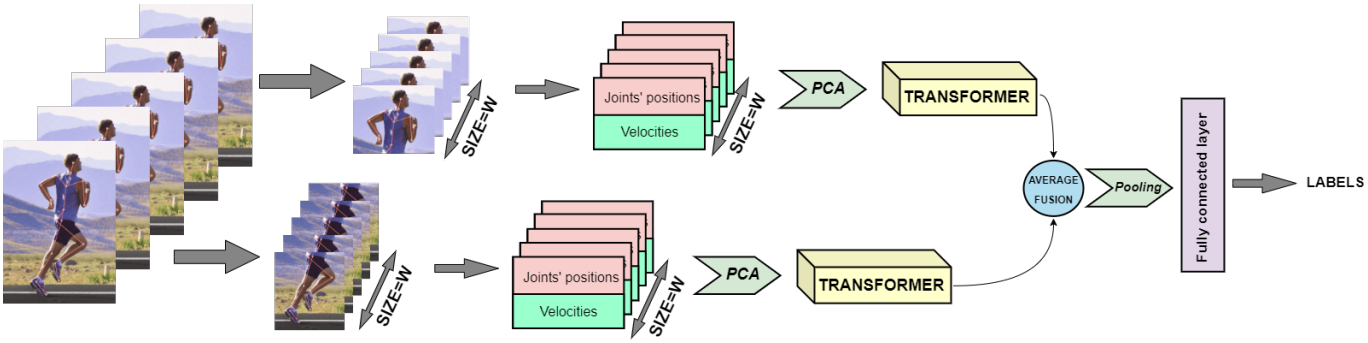


Fig. 2. Architecture of proposed method.

Next, we **normalized all action-snippets** using average height and position of neck:

$$\hat{s}^t = ([\xi_1^t, \xi_2^t, \xi_3^t, \ldots, \xi_{|\delta|}^t] - \xi_0^t)/\hat{H} \quad (3)$$

$$p_{joints} = [\hat{s}^1, \hat{s}^2, \hat{s}^3, \ldots, \hat{s}^W] \quad (4)$$

Where subtraction of $\xi_0^t$ and division by $H$ means element-wise subtraction and division.

By using normalized joint positions we calculated **joint velocities** ($v_{joints}$):

$$v_{joints} = [(\hat{s}^2 - \hat{s}^1), (\hat{s}^3 - \hat{s}^2), \ldots, (\hat{s}^W - \hat{s}^{W-1})] \quad (5)$$

The above subtractions are performed element-wise.

After, we separately calculated **velocity of neck** ($v_{neck}$), since its solo value is as much informative as other features:

$$v_{neck} = [(\xi_0^2 - \xi_0^1), (\xi_0^3 - \xi_0^2), \ldots, (\xi_0^W - \xi_0^{W-1})] \quad (6)$$

Here, $v_{neck}$ is repeated 10 times to increase its significance. After these computations, joint-based features are flattened and concatenated for further processes:

$$\mathbf{f} = [p_{joints}, v_{joints}, v_{neck}] \quad (7)$$

### D. Transformer model

In our proposed approach, we used a two-branch Transformer to process joint-based features of upper and lower body parts separately. These two branches of Transformer have identical architecture and consist of a positional embedding layer, encoder block, global max-pooling layer, and fully connected layer. The positional embedding layer is used to capture order information since action-snippets, fed into the Transformer, are the ordered combination of joint-based features of consecutive frames. Meanwhile, the encoder block in the transformer consists of multi-head attention and feed-forward layers.

Upper and lower body parts' vectors of joint-based features ($\mathbf{f}$) in (7) are reduced in dimension by PCA and fed through these two branches of Transformer. After, their outputs are combined by using the average fusion method, which is defined below:

$$\mathbf{f}^{average} = (\mathbf{f}^{up} + \mathbf{f}^{low})/2 \quad (8)$$

Where addition and division operators are executed element-wise. Then, this combination is processed through global max-pooling and fully connected layers to make classifications.

## IV. Experimental results

### A. Dataset

Our model was trained and tested on an open-source dataset called **JHMDB**. This dataset consists of 928 videos with 3 different splits into training and testing sets. These videos are arranged into 21 action labels, each video containing 32 frames. Moreover, in this dataset, authors manually annotated the 2D skeletons of people, and each skeleton consists of 15 joint positions.

### B. Experimental setup

In our work, we trained the Transformer model without any hyperparameter-tuning. Values of the hyperparameters were chosen in accordance with [6]. Specifically, we set the window size $W$ to 5, and the number of dimensions after applying PCA is 100. Our model was trained for 30 epochs. Results and numbers are based on the average accuracy of 3 different splits of JHMDB.

### C. Results

*1) HAR accuracy with different combinations of features:* Firstly, we experimented with different combinations of features. We tried to add distance feature [6], which is a Euclidean distance between joints in upper and lower parts of each skeleton in an action-snippet, to feature vector ($\mathbf{f}$) in (7). We found that the combination of velocity and position features is superior to others. Table I shows these results. Note that outputs of the two branches of Transformer were just concatenated.

*2) HAR accuracy with different fusion methods:* Secondly, we tried to apply different fusion methods to the outputs of the two parallel branches of Transformer. Using fusion methods could help us to derive more informative feature vectors. We placed fusion methods immediately after the global max pooling layer. As shown in Table II, the average fusion method had the highest accuracy. However, replacing this fusion method before the pooling layer increased the accuracy of the model.

TABLE I
ACCURACY OF DIFFERENT FEATURE COMBINATIONS ON JHMDB

| Feature combination | Accuracy |
|---|---|
| Velocity+Distance [6] | 71.19% |
| Position+Distance [6] | 65.81% |
| Position+Velocity+Distance [6] | 71.8% |
| Position+Velocity | **75.19%** |

TABLE II
ACCURACY OF FUSION METHODS ON JHMDB

| Fusion method | Accuracy |
|---|---|
| Concatenation | 75.19% |
| Sum | 75.82% |
| Multiplication | 60.52% |
| Average | 76.95% |
| Average (before pooling layer) | **77.95%** |

*3) Comparisons with state-of-the-arts:* We compared our model with prior state-of-the-art frameworks. We also tried different model types to train. Specifically, a single Transformer model, where upper and lower joint-based features are concatenated before processing through Transformer, and a two-branch Transformer without PCA. It is shown in Table III that our proposed method outperforms all the given models in terms of accuracy. It is also worth noting that our model has the least size (i.e., number of learned parameters) after EHPI [9].

TABLE III
COMPARISON WITH THE STATE-OF-THE-ARTS ON JHMDB

| Method | Model Size | Accuracy |
|---|---|---|
| ChainedNet [8] | 17.5M | 56.8% |
| EHPI [9] | 1.22M | 65.2% |
| Potion [10] | 4.87M | 67.9% |
| JMRN [11] | - | 68.55% |
| DD-Net [12] | 1.82M | 77.2% |
| Transformer [6] | 1.45M | 74.7% |
| Two-branch Transformer | 1.34M | **77.95%** |
| Two-branch Transformer w/o PCA | 6.16M | 70.47% |
| Single Transformer | 2.66M | 76.71% |

## V. CONCLUSION

In this paper, we proposed a model based on a two-branch Transformer. It was shown that treating human upper and lower body parts separately and processing features of each body part through Transformer has high accuracy compared to state-of-the-art frameworks. We should also note that our model did not go through any hyperparameter-tuning processes. Our future work will consider other features, such as the angle between joints, the distance between joints of two body parts, and their combination with features defined in this paper. We will also focus on hyperparameters and try to update our model.

## VI. ACKNOWLEDGMENTS

This work was supported by Faculty Development Competitive Research Grant Program (No. 11022021FD2925) at Nazarbayev University.

## REFERENCES

[1] N.A. Tu, K.S. Wong, M.F. Demirci, and Y.K. Lee, "Toward efficient and intelligent video analytics with visual privacy protection for large-scale surveillance". *The Journal of Supercomputing*, pp.14374-14404, 2021.
[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. and Polosukhin, "Attention is all you need." *Advances in neural information processing systems*, 2017.
[3] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Vision-based human action recognition: An overview and real world challenges," *Forensic Science International: Digital Investigation*, p. 200901, 2020.
[4] K. Schindler and L. Van Gool, "Action snippets: How many frames does human action recognition require?" in CVPR. IEEE, 2008, pp. 1–8.
[5] C. Wang, J. Flynn, Y. Wang, and A. Yuille, "Recognizing actions in 3d using action-snippets and activated simplices," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[6] A. Askar, L. Min-Ho, T. Huynh-The, and N. A. Tu "2D skeleton-based Human Action Recognition using Action-Snippet Representation and Deep Sequential Neural Network", IEEE SMC 2022.
[7] D. Avola, M. Cascio, L. Cinque, G. L. Foresti, C. Massaroni and E. Rodolà, "2-D Skeleton-Based Action Recognition via Two-Branch Stacked LSTM-RNNs," in *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2481-2496, Oct. 2020
[8] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *IEEE ICCV*, 2017, pp. 2904–2913.
[9] D. Ludl, T. Gulde, and C. Curio, "Simple yet efficient real-time pose-based action recognition," in *Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 581–588.
[10] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "Potion: Pose motion representation for action recognition," in *IEEE CVPR*, 2018, pp. 7024–7033.
[11] A. Shah, S. Mishra, A. Bansal, J.-C. Chen, R. Chellappa, and A. Shrivastava, "Pose and joint-aware action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3850–3860.
[12] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, "Make skeleton-based action recognition model smaller, faster and better," in *Proceedings of the ACM multimedia asia*, 2019, pp. 1–6.
[13] Y. Han, S.-L. Chung, Q. Xiao, W. Y. Lin, and S.-F. Su, "Global spatio-temporal attention for action recognition based on 3d human skeleton data," *IEEE Access*, vol. 8, pp. 88 604–88 616, 2020.
[14] T. Huynh-The, C.-H. Hua, N. A. Tu, and D.-S. Kim, "Learning geometric features with dual–stream cnn for 3d action recognition," in *ICASSP*. IEEE, 2020, pp. 2353–2357.
[15] S. Wei, Y. Song, and Y. Zhang, "Human skeleton tree recurrent neural network with joint relative motion feature for skeleton based action recognition," in *ICIP*. IEEE, 2017, pp. 91–95.
[16] F. Angelini, Z. Fu, Y. Long, L. Shao, and S. M. Naqvi, "2d pose-based real-time human action recognition with occlusion-handling," *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1433–1446, 2019.
[17] X. Jiang, K. Xu, and T. Sun, "Action recognition scheme based on skeleton representation with ds-lstm network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2129–2140, 2019.
[18] Duan, Haodong, et al. "Revisiting skeleton-based action recognition." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.