Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.34575

Lightweight Combat Motion Recognition Based on Human Keypoint Extraction by MediaPipe Models

Guanlin Guo[1*], Jinyu Ye[2], Taiyi Yu[3]

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
[2]St. Francis Xavier's School, Hong Kong, China
[3]School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei, China
**Corresponding Author. Email: Guanlin.Guo23@student.xjtlu.edu.cn*

Abstract. With the development of computer vision technology, motion recognition based on key points in the human body has also found many promising applications in sports analysis, intelligent monitoring and human-computer interaction. This paper proposes a lightweight motion recognition method for combat actions. The first is the MediaPipe framework for extracting human keypoints. Build a Fusion-Score Algorithm Next. Systematically calculate multi-dimensional features of the distance, speed and joint angle to detect punch motion. Two groups of comparative experiments have been conducted to assess the performance of the feature fusion method. Based on the above experiments, the multi-feature fusion algorithm is better than those employing only one feature in terms of both accuracy and robustness, with a success rate of 50%. Experimental results show that the proposed multi-feature fusion algorithm outperforms single-feature methods in both accuracy and robustness, verifying the effectiveness and application potential of the proposed method in real-world action recognition.

Keywords: Human Key Points, Motion Recognition, Lightweight, Multi-feature Fusion, MediaPipe

1. Introduction

1.1. Research background

A typical type of computer vision is the recognition of human behaviour. In short, it is to make computers understand what people are doing in videos and pictures. In recent years, deep learning technology has become more and more powerful, and now many applications of recognition technology are being used in areas such as intelligent surveillance, sports analysis, virtual reality, medical rehabilitation, and so on [1, 2]. Accurately identify punches and kicks in the analysis of combat sports to help evaluate the progress of training for athletes, learn about game situations, and offer intelligent fitness guidance.

In the past, for action recognition, have primarily used RGB videos and applied convolutional or recurrent neural networks for feature extraction to simplify the classification task [3, 4]. However, these methods have a serious problem: they are computationally intensive and therefore require a

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.34575

large amount of labelled data. They are also very easily distracted by their surroundings, so a messy or dim place may not be convenient for them to see. Now it is different. Recognition based on the skeleton model of the human body is now relatively popular. Since this way is less affected by changes in the environment, it requires less data and is significantly faster to process [5, 6].

1.2. Existing problems

In recent years, some methods for detecting changes in the human skeleton have achieved good results. However, there are still many problems that have not been solved.

First, the amount of calculation is very large. Although the high-recognition-accuracy deep learning methods ST-GCN and MS-AAGCN are available now, they have too many model parameters and are computationally expensive. Therefore, they are not suitable for the requirements of a real-time application.

Secondly, deep learning generally needs a large amount of labelled data for training. Datasets tailored for combat operations are available but scarce. Thus, the models trained on this data will have a generalisation ability to a certain extent.

The third reason is that cannot explain why this thing works in this way. Deep learning models are black boxes;and don't know why they have arrived at that conclusion. Some places that are hard to learn about are generally complex, such as the systems helping to make decisions by referees.

1.3. Contributions of this paper

In this paper, I have proposed a relatively simple combat action recognition method. What are the merits of this way? Let me tell you some of them.

I have proposed a new concept in this paper: " Rather than using a complex deep learning model, various key points in the human body are geometrically related, and thus rules for action recognition have been designed. It will be relatively low in cost.

A Multi-Feature Fusion Model for Action Scoring Has Been Constructed. The above are all used in combination to make judgments about the change in distance, speed and angles of movement, etc. Then, the above features are combined by a weighted fusion method to form a single action-scoring function.

This paper developed a one-stop solution for identifying actions in videos directly. MediaPipe will be used to get the key points and features, as well as recognize gestures at the same time. As a result, the system has been performing well recently and is quite scalable.

2. Relevant works

2.1. Methods for human pose estimation

Human pose estimation is a technology that finds the locations of key points on people in pictures and videos, such as joints and heads, and is used to support action recognition. Based on the output dimension, it can be divided into 2D pose estimation and 3D pose estimation; according to the number of people being processed, it can be further divided into single-person pose estimation and multi-person pose estimation [7, 8].

Top-down way: First, find the bounding box of a person, and then locate the corresponding keypoints inside this box. Representative works are CPN [8], SimpleBaseline [9] and HRNet [10]. CPN addresses the "difficult" keypoints well by a two-stage structure of GlobalNet and RefineNet,

DOI: 10.54254/2755-2721/2026.34575

and HRNet maintains high-resolution representations at all times through a high-resolution network architecture.

Bottom-Up approach: Directly detects all keypoints and then clusters or groups them according to different human bodies.

OpenPose [11] introduced Part Affinity Fields (PAFs) to encode the associations of keypoints; HighHRNet [12] addressed the problem of scale variation with a high-resolution feature pyramid and achieved state-of-the-art performance on the COCO dataset at that time.

Lightweight Approach: Given the demands of mobile devices and real-time applications, Google's BlazePose [13] is an encoder-decoder structure that achieves over 30 frames per second on Pixel 2 smartphones and provides a feasible solution for real-time pose estimation.

2.2. Skeleton action recognition methods

The categories of skeleton-based action recognition methods based on feature extraction are as follows:

RNN-based methods: Treat skeleton sequences as time series of joint coordinates and model temporal dependencies with LSTM or GRU. Du et al. [14] put forward a hierarchical bidirectional RNN that divides the human body into several sections for individual handling; Zhang et al. [15] added a perspective transformation mechanism to convert skeleton data into a more suitable view automatically.

CNN-based methods: Convert skeleton data into pseudo-images and use the excellent spatial feature extraction abilities of CNNs. Kim and others [16] directly concatenated the joint coordinates into a one-dimensional vector for input into a residual CNN; Liu and others [17] designed ten spatio-temporal image encoding methods to enhance the expressive power of skeleton data.

GCN-based methods: Model skeleton data as a graph and use graph convolutional networks to extract topological relationships among joints. ST-GCN [18] was the first to use GCN for skeleton action recognition and modeled joint dependencies through spatio-temporal graph convolutions; MS-AAGCN [19] further introduced adaptive graph convolutional layers that could dynamically adjust the graph structure based on input data to improve recognition performance significantly.

Transformer-based methods: Recently, the Transformer architecture has also shown good results in action recognition. Zhang et al. [20] considered skeleton sequences as token sequences and used a self-attention mechanism to learn long-range dependencies.

2.3. Lightweight action recognition

Lightweight action recognition is simply to make the model simpler and reduce the computational burden so that it can run on edge devices with limited resources. How exactly is it achieved? The following are the main ways:

Model compression: Reduce the size of the pre-trained model using knowledge distillation, pruning and quantization. Andronov's group constructed a Video Transform Network (VTN) to "extract" knowledge from multimodal models and then converted it into a lightweight single-modal model. Therefore, this model can be executed on a CPU for real-time processing at a speed of 56 frames per second.

Good architecture design: Need to build a light-weight network structure, such as MobileNet and ShuffleNet. MediaPipe [16] has optimised both the network structure and the inference process to achieve high accuracy and is relatively fast on mobile devices for real-time processing.

DOI: 10.54254/2755-2721/2026.34575

Rule-based lightweight methods: These do not use data-driven deep learning models but rather employ manually designed rules and thresholds to determine behaviour. Their generalisation capability is indeed relatively weak, but they have some advantages in specific situations: they are simple to implement, highly interpretable and computationally inexpensive.

3. Methods

3.1. General system framework

Figure 1 shows the lightweight fighting action recognition process used in this experiment. In the process, fighting video frames are used as the input, and the coordinates of human key points are extracted by MediaPipe Pose. After these key points are obtained, the shoulder, elbow and wrist points related to punching are selected; then distance change, wrist speed and elbow angle are calculated based on these upper limb points. These features are combined by a weighted score, and the score is compared with a preset threshold, so the system can judge whether the current action is a punch. The whole process does not use complex model training, but mainly depends on key point coordinates and simple geometric features to complete action recognition.

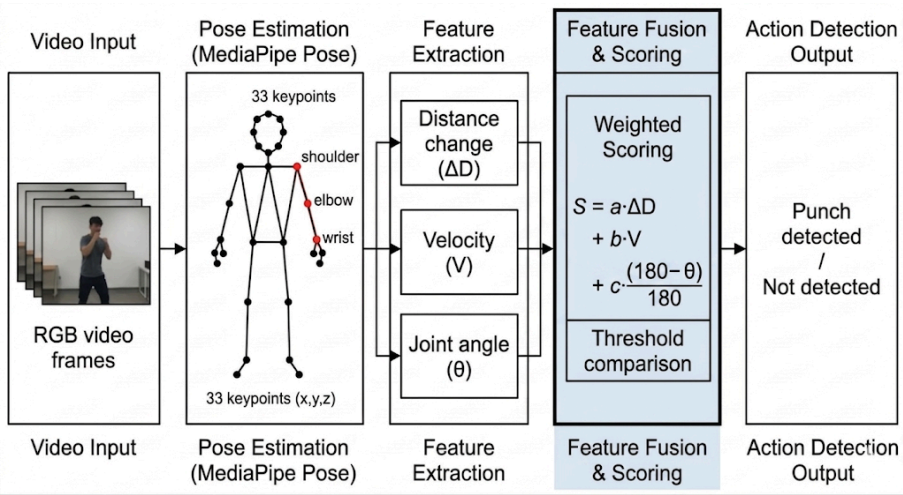

Figure 1. Process of fighting action recognition based on key points (picture credit: original)

3.2. Extraction of human body keypoints

MediaPipe Pose was used to find the position of people in a video in an experiment. The three dozen key points of the body detected by the model are shoulders, elbows and wrists. As punching movements are mainly extensions and retractions of the upper limbs, this study will focus on analysing the shoulders, elbows and wrists. The output of MediaPipe in this experiment includes the x and y coordinates as well as a relative depth z, but the two-dimensional feature calculation mainly uses only the x and y coordinates.

3.3. Feature design

This article selects distance variation, motion speed, and joint angle as the main features for recognizing punching movements, represented by ΔD, V, and θ, respectively. Among them, ΔD represents the change in distance between the shoulder keypoints and the wrist keypoints, which is used to reflect the spatial position changes caused by the forward extension of the arm during the

DOI: 10.54254/2755-2721/2026.34575

punching process; V is calculated from the displacement of wrist keypoints in adjacent video frames, and is used to characterize the speed of hand movement; θ is calculated based on the geometric relationship between the three key points of the shoulder, elbow, and wrist to describe the degree of bending and extension of the elbow joint. Due to the fact that punching movements are usually accompanied by rapid arm extension and elbow joint extension, the above three types of features can provide a basis for punching judgment from the perspectives of displacement, velocity, and posture changes.

3.4. Multi-feature fusion scoring function

A single-feature judgment is more susceptible to interference from variations in action amplitude, speed fluctuations and key-point jitter. To improve the stability of recognition, distance changes, wrist velocity and elbow joint angles are weighted-combined to form an overall score S in this paper. If S is greater than the preset threshold, then it is determined that there has been a punching operation in the neighbourhood of the current frame.

$$S = a \cdot \Delta D + b \cdot V + c \cdot \frac{180-\theta}{180} \tag{1}$$

Among them, a, b, and c are the weight coefficients of the three types of features respectively; ΔD represents the change in distance; V represents wrist speed; θ is the elbow joint angle, and the angle term is used to describe the degree to which the arm approaches extension. To select the combination of weights and thresholds with better recognition performance, this paper adopts grid search method for parameter optimization within the candidate parameter range.

4. Experiment

4.1. Experimental environment

The operating system in this experiment is Windows 11, and Python 3.10.11 was used for the implementation. The main dependent libraries are OpenCV 4.13.0, MediaPipe 0.10.9, NumPy 2.2.6 and Pandas 2.2.3. The hardware platform is an Intel Core i7-12700H processor, and GPU acceleration is not used.

4.2. Dataset and preprocessing

The data for this study are publicly available combat videos; five sets of video clips with 23 punching movements were selected, and manually annotated keyframes were used as the ground truth. MediaPipe Pose is used to extract frame-by-frame coordinates during processing and saved in a CSV file; analysis of the two-dimensional data for upper limb joints (shoulders, elbows, wrists) is conducted. Then, the system obtains displacement, velocity and angle features for comparison. For the above statistical data to be accurate, a minimum frame interval constraint has been set to prevent multiple triggers of the same action.

DOI: 10.54254/2755-2721/2026.34575

4.3. Experimental design

Because the impact of distance change, speed and joint angle on punching determination varies to different extents, weights a, b, c and the determination threshold in Method 3 need to be determined through experiments. A grid search method is used in this paper to test all the combinations of weights and thresholds within a certain range of candidate parameters, and then, for each combination, the corresponding Precision, Recall and F1 scores are calculated. Considering both the accuracy of detection and the number of false positives, the parameter combination with a relatively high F1 score and fewer false positives was finally selected as the final experimental parameters for Method 3.

4.4. Experimental results

Figure 2 shows the F1 score comparison results of the three methods for each action group with a matching window of ± 5 frames. As shown in the figure, Method 3 performed relatively well in the G4 and G5 groups, and the general trend of change was relatively stable. Figure 3 is the experimental result after reducing the matching window to ± 3 frames. Compared with the ± 5 frame window, the F1 score fluctuations of all methods are more significant under the ± 3 frame condition, and thus the size of the matching window will affect the evaluation results of action detection.

Table 1 presents the F1 scores of the three methods for different action groups, and Table 2 shows other indicators such as mean precision, mean recall, mean F1 score, standard deviation of F1 score, and mean FPR. Based on the experimental results, with a ±5 frame matching window, the average F1 score of method 3 is 0.67; this is higher than that of method 1 (0.60) and method 2 (0.63). Therefore, it can be seen that after introducing multi-feature fusion, the entire detection performance of the system has been improved.

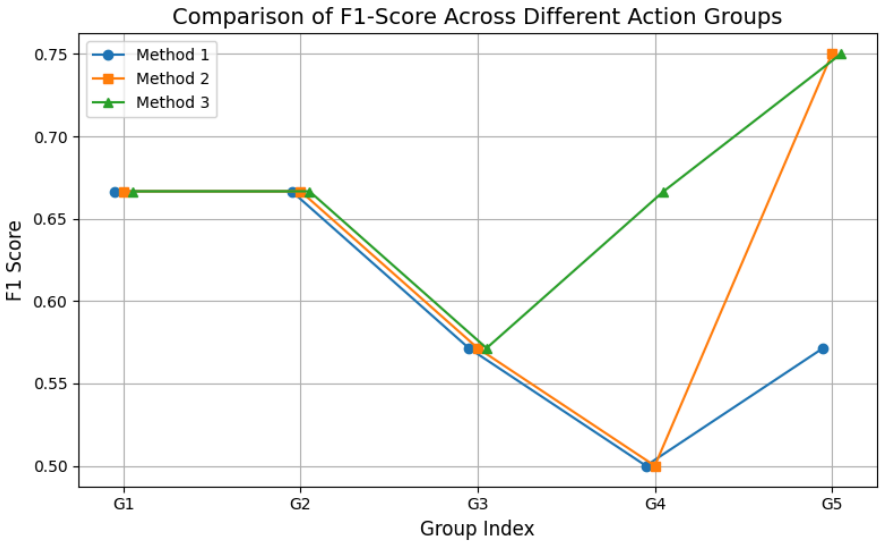

Figure 2. F1 score comparison of different methods for each action group (matching window ± 5 frames)(picture credit: original)

DOI: 10.54254/2755-2721/2026.34575

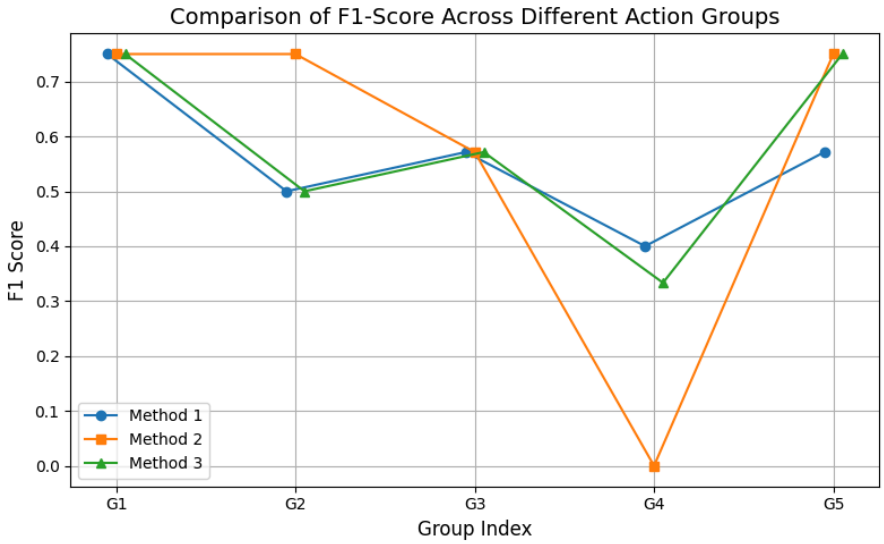

Figure 3. F1 score comparison of different methods for each action group (matching window ±3 frames)(picture credit: original)

Table 1. F1 score of different methods for each group (matching window ± 5 frames)

Grouping	Method 1 F1	Method 2 F1	Method 3 F1
G1	0.67	0.67	0.67
G2	0.67	0.67	0.67
G3	0.57	0.57	0.57
G4	0.50	0.50	0.67
G5	0.57	0.75	0.75

Table 2. Summary of average performance for different methods (matching window ± 5 frames)

Method	average precision	average recall	average F1	F1 standard deviation	average FPR
Method 1	1.00	0.43	0.60	0.07	0.00
Method 2	1.00	0.47	0.63	0.09	0.00
Method 3	1.00	0.50	0.67	0.06	0.00

4.5. Result analysis

Based on the results of the experiment, method 3 showed relatively good performance, and it can be seen that only a single feature cannot fully represent boxing movements. Method 1 is mainly based on the distance change for judgment, and it is easily affected by differences in action amplitude and key point jitter; Method 2 introduces speed features on this basis. Although the F1 score increased for some action groups, the results were highly variable and unstable. Method 3 describes the punching process in three ways based on information such as distance change, wrist velocity and joint angle to achieve relatively stable overall results.

The Size of the matching window will also affect the final evaluation results. In the experiment, when the matching window was changed from ±5 frames to ±3 frames, the results of the three

DOI: 10.54254/2755-2721/2026.34575

methods were all reduced to some extent; this indicates that a narrower range for frame matching will make the detection results less likely to align with the manual labels. One reason is that the manually marked punch frame and the frame predicted by the program are not exactly the same. Therefore, in a smaller window, some predictions that are close to the actual punch moment may still be considered missed detections. Therefore, the ±5 frame window is selected as the primary evaluation case in this paper, and the ±3 frame window is also kept as a comparative option.

5. Conclusions

This paper has proposed a lightweight method for recognizing punch actions. MediaPipe is used to obtain body keypoints in this manner, and then three motion features are extracted from them: distance change, wrist velocity, and elbow joint angle; finally, a fusion scoring function for punch detection is constructed. Based on the experiment, the multi-feature fusion method is better than the other two in all aspects and is therefore feasible. The way that is shown in this paper is relatively simple and convenient; therefore, it is very suitable for applications with high real-time requirements and low-power computers. However, the present method does not have good generalisation for all kinds of combat styles and motions. To address the above problem, the next set of research will extend the scope of range-of-motion recognition to include kicks and defensive postures. Machine Learning Technology can be used to improve Recognition accuracy.

Author contribution

All the authors have contributed equally and have been listed alphabetically by name.

References

[1] Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., ... & Shah, M. (2023). Deep learning-based human pose estimation: A survey. ACM computing surveys, 56(1), 1-37.
[2] Ren, B., Liu, M., Ding, R., & Liu, H. (2024). A survey on 3d skeleton-based action recognition using learning method. Cyborg and Bionic Systems, 5, 0100.
[3] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems, 27.
[4] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).
[5] Liu, J., Shahroudy, A., Xu, D., Kot, A. C., & Wang, G. (2017). Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. IEEE transactions on pattern analysis and machine intelligence, 40(12), 3007-3021.
[6] Zhang, S., Liu, X., & Xiao, J. (2017, March). On geometric features for skeleton-based action recognition using multilayer lstm networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 148-157). IEEE.
[7] Liu, W., Bao, Q., Sun, Y., & Mei, T. (2022). Recent advances of monocular 2D and 3D human pose estimation: A deep learning perspective. ACM Computing Surveys, 55(4), 1-41.
[8] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., & Sun, J. (2018). Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7103-7112).
[9] Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5693-5703).
[10] Xiao, B., Wu, H., & Wei, Y. (2018). Simple baselines for human pose estimation and tracking. In Proceedings of the European conference on computer vision (ECCV) (pp. 466-481).
[11] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7291-7299).

DOI: 10.54254/2755-2721/2026.34575

[12] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., & Zhang, L. (2020). Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5386-5395).
[13] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv: 2006.10204.
[14] Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1110-1118).
[15] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In Proceedings of the IEEE international conference on computer vision (pp. 2117-2126).
[16] Soo Kim, T., & Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 20-28).
[17] Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition, 68, 346-362.
[18] Yan, S., Xiong, Y., & Lin, D. (2018, April). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).
[19] Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2020). Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Transactions on Image Processing, 29, 9532-9545.
[20] Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., & Zheng, N. (2020). Semantics-guided neural networks for efficient skeleton-based human action recognition. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1112-1121).