Int. J. Simul. Multidisci. Des. Optim. **15**, 9 (2024)

https://doi.org/10.1051/smdo/2024004

IJSMDO
Available online at:
https://www.ijsmdo.org

RESEARCH ARTICLE OPEN ACCESS

Application of 3D recognition algorithm based on spatio-temporal graph convolutional network in basketball pose estimation

Mingzhi Ye*

Graduate School, University of Perpetual Help System DALTA, Manila City 1740, Philippines

Received: 21 August 2023 / Accepted: 18 March 2024

Abstract. In recent years, human motion recognition in computer vision has become a hot research direction in this field. Based on 2D human motion recognition technology, real-time extraction of motion features from 2D planes is used to recognize human movements. This method can only learn the position contour and color information of the image. It cannot directly reflect the motion situation, which results in low recognition accuracy and efficiency. In response to this issue, this study proposes a combination method of motion recognition and 3D pose estimation to recognize and classify basketball movements. First, the 2D skeleton model is obtained by extracting the feature information in the video action, which is converted into a 3D model, and the model is replaced by the time-space convolutional network to establish a human action recognition model. The experiment showed that when the number of iterations reached 6, the accuracy of the spatio-temporal graph convolutional network algorithm model reached 92%. Comparing the accuracy rates of different algorithm models, the average accuracy rates of convolutional neural network, long short memory network, graph convolution, long short model of action recognition and graph convolution model of action recognition were 61.6%, 65.4%, 72.5%, 76.8% and 90.3% respectively. The results show that the proposed 3D recognition algorithm can accurately recognize different basketball movements. This study can provide reference for basketball coaches and athletes in basketball training.

Keywords: Spatio-temporal graph / CNN / video analysis / action recognition / attitude estimation / skeleton model / AI / modelling

1 Introduction

The internet has led to the high-speed growth of Artificial Intelligence (AI) technology in various fields, and various new technologies combined with deep learning have become the research focus of AI [1]. In the field of computer vision, AI technology is used to connect computers and cameras to recognize and classify images in videos. Human Pose Estimation (HPE) is the process of extracting feature information from images and comparing the extracted information with the information present in the database to determine the actions of various parts of the human body [2]. This study uses the Regional Multi-person Pose Estimation (RMPE) algorithm to estimate human posture. However, due to the shortcomings of this algorithm, Spatial Transformer Networks (STN) are introduced as auxiliary algorithms for HPE. Then, the data set is transformed by building a Human Skeleton Model (HSM). Finally, the transformed data is recognized and actions are classified by Spatio-temporal Graph Convolutional Networks (ST-GCN). The content mainly has four parts. The first part briefly describes the research topics related to human body recognition by other scholars. The second part is a review of the main methods used in this study. The third part is the analysis of the model results and model data obtained through research methods. The fourth part is a summary of all the above content and prospects for future research.

2 Related work

With the great breakthroughs in internet technology, AI has been applied in various fields. Cui et al. found that traffic prediction is a particularly challenging application module for spatio-temporal prediction. For this module, the team proposed a new deep learning framework, which uses ST-GCN combined with Long Short-Term Memory network (LSTM) to predict the traffic status of the whole network. On two different real-world traffic state datasets, the ST-GCN model outperformed the conventional baseline method [3]. Yang et al. proposed a new fine-grained method for predicting urban crowd flow in public

* e-mail: ymingzhi@outlook.com

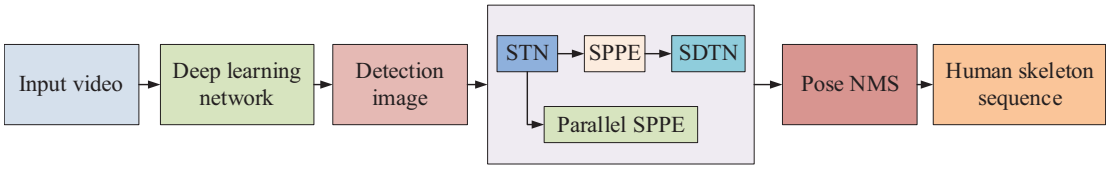


Fig. 1. Overall structure of RMPE algorithm.

places. This method was based on an end-to-end structure of adaptive ST-GCN with spatio-temporal data characteristics to predict pedestrian flow in simultaneous inflow, outflow, and flow directions, and established a crowd flow model. This method using ST-GCN was superior to other existing methods, with smaller errors and higher accuracy [4]. Zhou et al. found that in the current video surveillance, autonomous vehicle and other fields, there is still a large development space for predicting the future pedestrian trajectory. Therefore, they proposed an ST-GCN based on attention interaction perception to predict pedestrian trajectories. This new prediction method had good performance in monitoring trajectories and calculating capture mechanisms [5].

Zhang et al. found that the accuracy of existing insect recognition and classification techniques in complex environments is not high. In response to this issue, they proposed a new morphological-based variable weight edge enhancement algorithm that uses narrowband fast methods to detect insect contours. This algorithm had higher network accuracy than traditional synthetic data algorithms and was more suitable for insect recognition in complex backgrounds [6]. Yang found that there is still some development space for image recognition algorithms in the field of traffic management. Therefore, researchers have proposed a vehicle recognition algorithm based on deep CNN. This algorithm recognized and classified input images to obtain more accurate vehicle information, which was more accurate than traditional algorithms in vehicle image classification and also provided better choices for intelligent traffic management [7]. Song et al. found that deep learning strategies used in facial recognition must be trained in a uniformly distributed large number of samples. To find a method for use in uneven environments, the team proposed a new recognition algorithm that marks the previously generated recognition data samples before forming a balanced training set. This algorithm had a high recognition rate and fewer over-fitting phenomena for faces [8].

In summary, many scholars have conducted research in the field of recognition and achieved significant results. However, most scholars have not applied ST-GCN to human body recognition when conducting research through recognition algorithms. ST-GCN, as a high-performance image analysis and recognition algorithm, can effectively recognize and predict human movements by combining HPE with ST-GCN. This study evaluates the performance of the model by detecting basketball movements in the dataset, and finally applies the model to basketball training, providing reference for basketball coaches and athletes' basketball training process.

3 Research on 3D recognition algorithm based on ST-GCN in motion and posture recognition

With the strong promotion of the Internet, human body recognition technology has also been widely applied in sports. This study proposes a Convolutional Neural Network (CNN)-based HPE model, which extracts data from images to establish a two-dimensional skeleton model (2D-SM), converts it into a three-dimensional skeleton model (3D-SM), and then uses ST-GCN to recognize human posture.

3.1 Study for CNN-HPE

HPE is a technology that uses images to locate and recognize human movements, thereby estimating the type of human movement. HPE can be divided into two estimation methods, i.e. 2D and 3D attitude estimation methods. This study estimates human posture through RMPE, which is an improved algorithm for estimating individual posture [9]. It can avoid the problems of inaccurate and complex detection box positions in single person estimation algorithms. Figure 1 shows the structure of the algorithm.

From Figure 1, the input video is used to detect the position of the human body in the image through a deep learning network, and the detected human frame is input into the Symmetric STN module to generate suggestions for human posture. The output attitude information is parameterized and inputted into the Pose NMS to obtain the Human skeleton sequence. In training, due to the inherent characteristics of the algorithm, it is easy to fall into local minima. Therefore, Parallel SPPE is introduced to improve this problem. This algorithm recognizes images through STN and parallel SPPE. SPPE can train individual images. The translation or deletion of attitude data by HPE can have a significant impact on the performance of SPPE [10]. In terms of human pose recognition, STN and SDTN have shown excellent performance. The 2D Affine transformation of STN is equation (1).

$$\begin{pmatrix} x_i^S \\ y_i^S \end{pmatrix} = [\theta_1 \theta_2 \theta_3] \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

In equation (1), x_i^S and y_i^S represent the coordinates of x_i^t and y_i^t before and after transformation. θ_1, θ_2 and θ_3 are vectors in 2D space. Human posture data is mapped to the

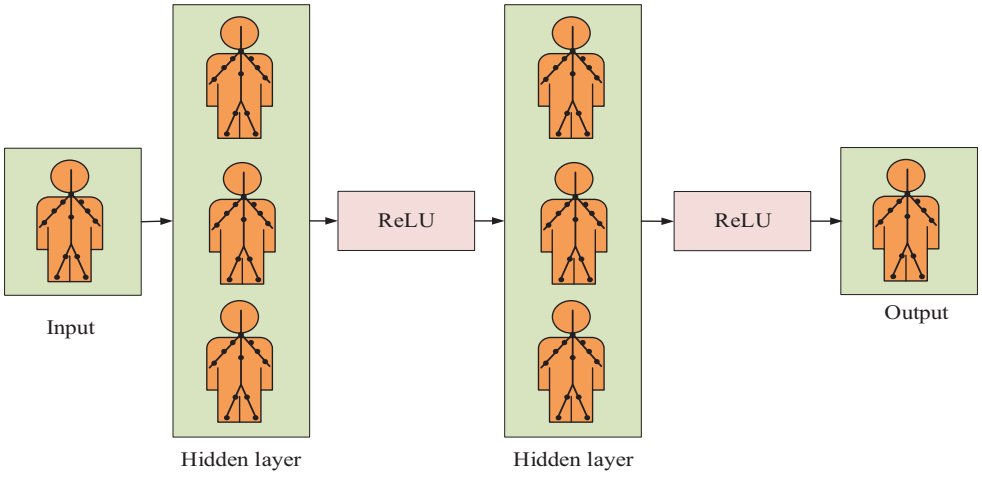


Fig. 2. Multi-layer GCN based on first-order filters.

initial human image through SPPE, and HPE is re-mapped using SDTN, as shown in equation (2).

$$\begin{pmatrix} x_i^t \\ y_i^t \end{pmatrix} = [\gamma_1\gamma_2\gamma_3] \begin{pmatrix} x_i^S \\ y_i^S \\ 1 \end{pmatrix} \tag{2}$$

In equation (2), γ represents the reciprocal of a 2D space vector. SDTN is the reversal process of STN, as shown in equation (3).

$$\begin{cases} [\gamma_1\gamma_2] = |\theta_1\theta_2|^{-1} \\ \gamma_3 = (-1)^*[\gamma_1\gamma_2]\theta_3 \end{cases} \tag{3}$$

In the model training stage, parallel SPPE can be viewed as a regularizer that can solve the local minimum problem during training, while also optimizing the problem of STN not changing the human posture to the center of the human body region [11]. Due to SDTN's ability to compensate for the model, errors in the network can be effectively reduced through this compensation. Under the action of parallel SPPE, STN is trained to move the human body to the recognition area, and posture estimation is performed on the data in the image through SPPE. In human pose detection, detectors are prone to generating redundant detection, resulting in low detection efficiency. Therefore, it is necessary to eliminate redundant detection and use the pose with the highest confidence as a reference. By comparing this posture with similar postures in the data, similar actions are eliminated in this posture.

3D pose estimation is the estimation of 3D pose in a video using a fully convolutional model based on extended time convolution at key points in 2D. This algorithm inputs 2D attitude data into the model through a time dilation convolution model, and uses time convolution for transformation. In transition mode, due to the fixed length of the gradient path between input and output, which is independent of the length of the array, this situation can lead to the disappearance of gradients and the minimization of gradient explosion problems in the convolutional network. In the case of limited 3D pose data, semi supervised training methods can be used to improve accuracy settings. By using video and 2D key point detection equipment, data loss can be effectively reduced and the recognition accuracy of the model can be enhanced. The main workflow of this method is that the encoder evaluates the 3D data points based on the 2D joint points, and then the decoder outputs the same 3D pose projection into the 2D coordinates. If the output data differs significantly from the original input, the training will be penalized [12]. The semi supervised training method combines supervised and unsupervised components together. The two objectives are optimized together, marking the location data with known data, then detecting unnamed data, and finally returning to the 2D space to check the consistency with the input.

With the development of artificial neural networks, Graph Convolutional Networks (GCNs) have been used for bone motion recognition. By using graph convolution to learn the spatial positional relationships between key bone nodes, actions are identified and classified. GCN is a network model specifically designed to handle graph structured data. In GCNs, each node represents an entity in the data, and each edge represents the relationship between entities. GCNs extract features by performing convolution operations on the graph, similar to applying convolution operations to extract features in an image. Specifically, GCNs update node representations by aggregating information from neighboring nodes of each node. This aggregation operation can be achieved by weighted averaging the features of neighboring nodes, which are usually obtained through learning. The main concept behind GCNs is to enhance the node representation by incorporating the graph's structural information. This ensures that the node representation not only contains its own feature information but also information about its neighboring nodes. Figure 2 shows the process of changing the image.

From Figure 2, the process of GCN changing images is an operation that passes the product between the image, image pixels, and kernel values through the kernel core. Traditional convolution involves selecting pixels around

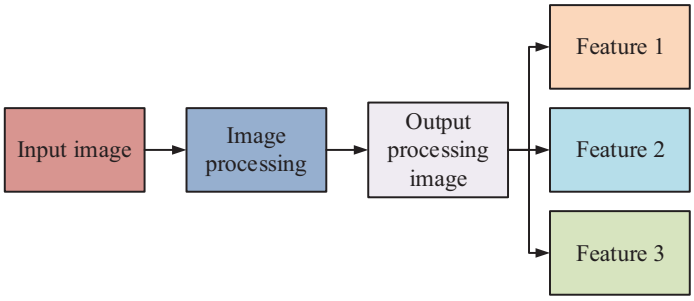


Fig. 3. GCN's inputs and outputs.

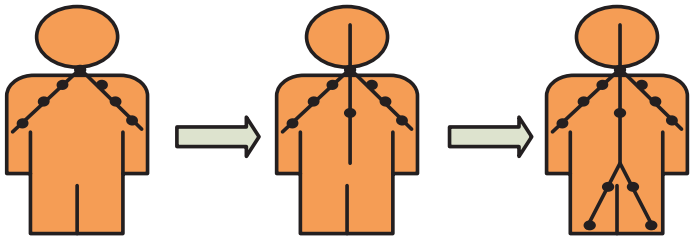


Fig. 4. Division of human skeleton.

the center and assigning weights to the pixels. However, there are differences between the pixels in bone data and other image data, and the placement of these pixels is irregular. Figure 3 shows the input and output process of GCN.

From Figure 3, image information is input into GCN, and node information is integrated through edge and corner information. Node information is created based on nodes, and then output to classify different information [13]. Single layer GCN is calculated by equation (4).

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{t}W^{l}) \tag{4}$$

In equation (4), W^t represents the untrained parameter. $\tilde{A}$ indicates that a self connected Adjacency matrix is added, and σ indicates the corresponding Activation function. $\tilde{D}$ represents the degree matrix. To classify the nodes of the two layer GCN, and their adjacent matrices are shown in equation (5).

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{5}$$

In equation (5), $\tilde{A}$ and $\tilde{D}$ are the same as equation (4). This result then input into GCN to obtain the predicted results for each label as shown in equation (6) [14].

$$Z = f(X, A) = soft\max(\tilde{A}ReLU(\tilde{A}XW^{o})W^{1}) \tag{6}$$

In equation (6), W^0 represents the position where the features of the node are mapped to the hidden layer, consisting of the first layer weight matrix of GCN. W^1 represents the hidden end of the node to the corresponding end of the hidden layer. *soft* max represents each node at the predicted end of each symbol. The expected entropy loss effect of all nodes is equation (7).

$$\xi = -\sum_{l\in Y_L}\sum_{f=1}^{F} Y_l \ln Z_{lf} \tag{7}$$

In equation (7), Y_L is the node set of the label. In the above methods, training models on special structures cannot be directly applied to different graph structures.

3.2 3D recognition algorithm based on ST-GCN

The action recognition used in this study is based on bone data, which identifies human bones in the video and extracts coordinate information for preprocessing. The extraction of 3D skeleton information requires extracting 2D skeleton information from the images in the video, and then transforming this information into 3D skeleton information [15]. The extraction of 2D skeleton information is divided into three steps, namely human frame detection, HPE, and non maximum suppression. The modeling of 2D human skeleton is Figure 4.

As shown in Figure 4, the image extracted from the video is first normalized, and then the graph is divided into multiple regions. Multiple priority boxes are used for each region grid, and the multidimensional attributes in the regions are found. Due to the randomness of region segmentation, there may be errors in the detected human frame. In response to this error, the detected human frame is input into two parallel pose estimation branches, and the initial detection frame is modified through STN, which can greatly improve the performance of SPPE evaluation. However, there will be a large number of repetitive actions in the detection box. To solve this problem, the attitude estimation results will be input into the Pose NMS module.

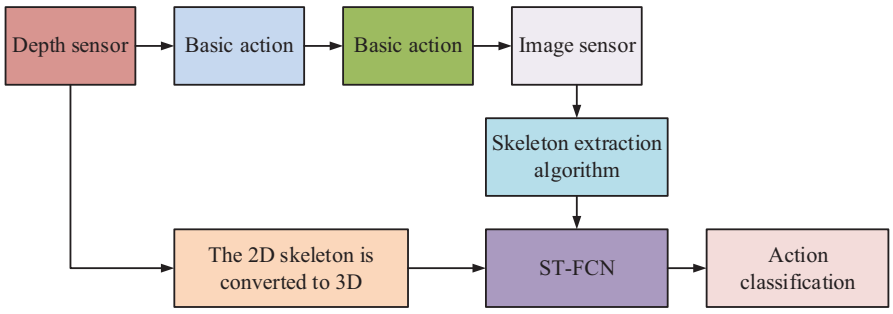


Fig. 5. Action recognition steps based on bone.

By comparing the human posture with the posture in the database, the coordinates of two-dimensional human key points are output to eliminate redundant issues in the detection frame [16]. Finally, the input 2D key points are processed using a fully convolutional network to obtain a sequence of 3D key points. The 2D coordinates are converted into 3D through multiple residual modules and void convolution factors. The ST-GCN model is used in this study, and Figure 5 shows the recognition process of the model.

As shown in Figure 5, the basic actions are input into the depth sensor and image sensor, and the skeleton data is extracted from the actions in the image through this sensor. The skeleton model is exported to ST-GCN, performing action classification on the skeleton data.

Traditional skeleton recognition uses convolution to recognize the movement of bones. During image recognition, the coordination of the human body determines the completion of an action, rather than a single joint point. Therefore, it is necessary to weigh the output feature map and set a convolutional kernel size of K. The output value in the channel is equation (8).

$$f_{out}(x) = \sum_{h=1}^{K}\sum_{\omega=1}^{K} f_{in}(p(x,h,\omega))W(h,\omega) \tag{8}$$

In equation (8), p is the sampling function centered on x, representing the neighborhood of x,w represents that the weight function calculates the inner product through the weight vector in three-dimensional space [17]. The sampling function is defined as an adjacent function relative to the center position. The weight function is equation (9).

$$W(v_{ti}, v_{tj}) = W'(l_{ti}(v_{tj})) \tag{9}$$

In equation (9), v_{ti} represents a node in the sampling function, and v_{tj} represents the sampling function of the path set between v_{ti} and v_{tj}. By improving the sampling function and weight function, a new weight function can be obtained as shown in equation (10).

$$f_{out}(v_{ti}) = \sum_{v_{tj}\in B_{v_{ti}}} \frac{1}{Z_{ti}(v_{tj})} f_{in}(p(v_{ti}, v_{tj}))\bullet W(v_{ti}, v_{tj}) \tag{10}$$

By normalizing the results, the final calculation formula is obtained by balancing different output subsets using this term, as shown in equation (11).

$$f_{out}(v_{ti}) = \sum_{v_{tj}\in B_{v_{ti}}} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{ti})\bullet W(l_{ti}(v_{tj})) \tag{11}$$

After completing feature weighting, dynamically to model the time and space. In the construction of skeleton mapping, the temporal aspect of the mapping is constructed by connecting the same joint points between consecutive frames. The temporarily connected joint points are shown in equation (12).

$$B(v_{ti}) = \left\{ v_{qj} | d(v_{tj}, v_{ti}) \le K, |q-t| \le \lfloor \tau/2 \rfloor \right\} \tag{12}$$

In equation (12), τ represents the size of the time kernel, which can control the time constraints included in the neighbor map [18]. The transformation of spatio-temporal maps also requires the participation of model functions and weight functions, as shown in equation (13).

$$l_{ST}(v_{qj}) = l_{ti}(v_{tj}) + (q - t + \lfloor \gamma/2 \rfloor \times K \tag{13}$$

In equation (13), l_{ti} represents the label of the application case, and the label mapping can be changed through the spatio-temporal environment. In this way, stable functions can be executed on structured spatio-temporal maps [19]. Figure 6 shows the specific process of ST-GCN.

In Figure 6, the process is divided into three parts. The first is to input the information data in the image, which is the coordinates of joint points in different images. Then, the edge weights of different parts of the human body are measured using the ST-GCN model [20]. Finally, the data is classified and the results are output through the average pooling layer and connection layer.

4 Performance analysis of 3D recognition algorithm based on ST-GCN in basketball motion estimation

This chapter first studies the accuracy of the algorithm model, determines the performance of the algorithm through different datasets, and then compares the operation time and iteration time of different algorithm models. Then it identifies different basketball movements to determine the performance of the model.

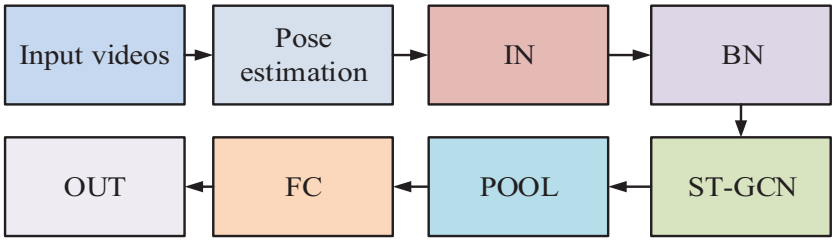
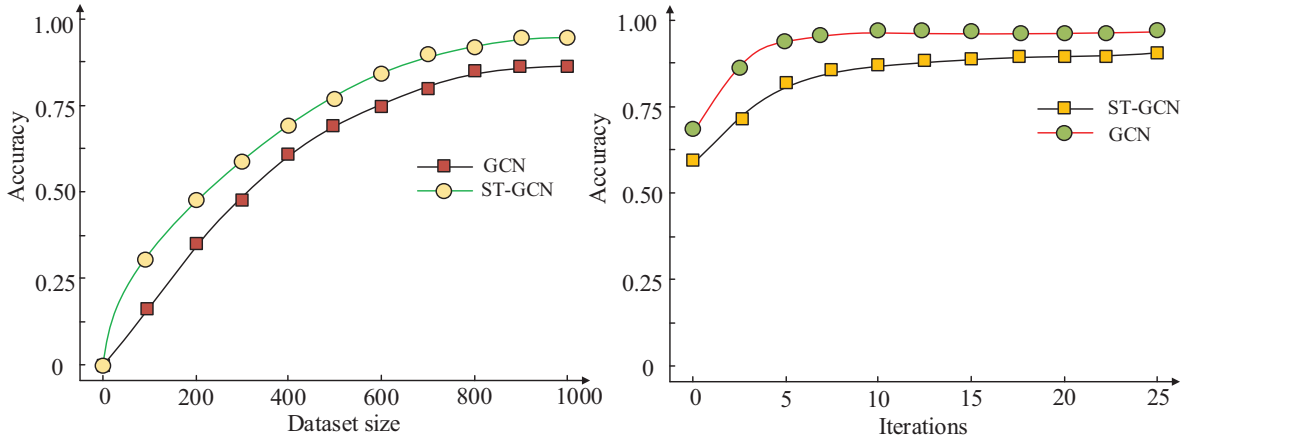


Fig. 6. Algorithm overall structure diagram.



(a)The relationship between dataset size and accuracy (b)The relationship between the number of iterations and accuracy

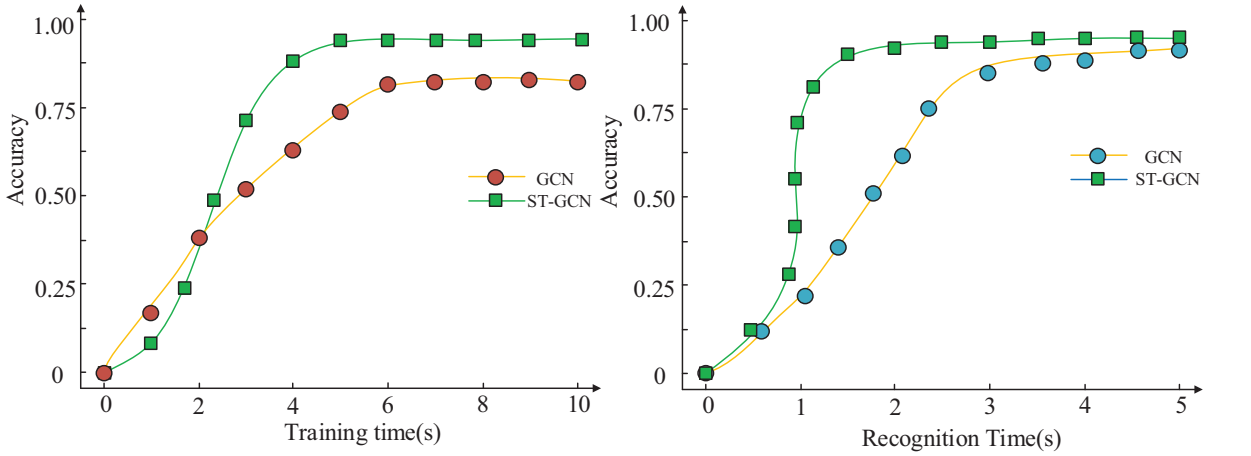
Fig. 7. Accuracy of two methods under different iteration times and dataset sizes.

4.1 Performance analysis of 3D recognition algorithm based on ST-GCN

To verify the performance of ST-GCN in 3D recognition, the GCN model is compared with ST-GCN. The CPU used in this experiment is Intel ® CoreTMi7-9700CPU@3.00GHz × 8. GPU is a server of NVIDIA GeForce RTX 2060 SUPER, using Windows 10 (64 bit) as the operating system. The UCF Basketball Data set is a widely used data set for basketball action recognition. It contains video clips of various basketball actions, including shooting, dribbling, passing, and defense, etc. Each video clip is labeled with a corresponding action category.

Figure 7a shows the relationship between dataset size and model accuracy, while Figure 7b shows the relationship between iteration times and model accuracy. The larger the training dataset for actions, the richer the action information it contains. The model can analyze different action information more accurately to improve the recognition accuracy of the model. When the dataset size of GCN and ST-GCN is 800, the accuracy of the trained model tends to stabilize, and the increase in accuracy with the increase of the dataset is very small. As the iteration increases, the accuracy of recognition also increases. When the ST-GCN iteration reaches 6 times, the model basically reaches its maximum accuracy and no longer increases with the increase of iteration. GCN achieves its best performance after approximately 10 iterations. The results show that the accuracy of the two algorithms in training sets with a size of 800 is 92.5% and 87.3%, respectively, and the accuracy of ST-GCN is higher than that of GCN in any dataset size. When the iteration reaches 6 times, the accuracy of ST-GCN reaches 92% and achieves the best performance. When reaching 10 times, the GCN accuracy reaches the optimal performance of 88%. When evaluating a model's performance, it is important to consider factors beyond just accuracy, such as training and recognition time. To achieve optimal performance for both models, the size of the training and validation sets is 1000 and the number of iterations is 25. Figure 8 illustrates the results of comparing the two methods.

Figures 8a and 8b show the relationship between the training time and recognition time of the two methods and the accuracy of the model. When the training time is limited, the accuracy of ST-GCN is lower, as this algorithm requires a large amount of data for training to improve model accuracy. When the training time reaches 2 seconds, the accuracy of ST-GCN greatly improves, and the performance is best when it reaches 5 seconds. ST-GCN can use a small amount of time to achieve high recognition accuracy in image recognition. The data shows that although the recognition accuracy of ST-GCN is slightly lower than that of GCN in less training time, as the training time increases, the model can achieve optimal performance in an extremely short time.



(a)The relationship between training time and accuracy (b)The relationship between recognition time and accuracy

Fig. 8. The relationship between training time and recognition time of two methods and accuracy.

4.2 Performance analysis of basketball motion recognition based on 3D recognition algorithm

To verify the performance of the 3D recognition model of ST-GCN, the performance of this method is validated under different dataset sizes, as exhibited in Figure 9.

As shown in Figure 9, as the dataset increases, the performance of the model gradually improves. When the dataset is small, the performance of ST-GCN is lower than that of ST-LSTM. However, when the dataset is increased to a certain extent, ST-GCN has a significant improvement and is more accurate than other methods. In basketball, certain movements are selected for recognition, including dribbling, backward jump shot, dribbling, dunking, and back up singles. These five actions are renamed as actions 1, 2, 3, 4, and 5. The impact of different parts of the human body on the accuracy of the recognition model is compared by adjusting different body weights, as listed in Figure 10.

Figure 10 a and 10b represent the recognition accuracy under different limb and torso weight coefficients, respectively. When the weight coefficients of the limbs change, the accuracy of the model increases with the increase of its coefficients. When the limb weight coefficient reaches 0.8, the accuracy of the model tends to stabilize. When the coefficient is constant, changing the weight coefficient of the body results in a smaller increase in model accuracy as the weight coefficient increases. The experimental data shows that the influence of trunk weight coefficients on model accuracy is smaller than that of limbs on accuracy. The detection of human movements by models mainly relies on the recognition of limb movements to determine human movements. The weight coefficients of different torso and limbs are combined, and then input into the model to recognize basketball movements, resulting in Figure 11.

As shown in Figure 11, the recognition accuracy of different actions under the third weight coefficient configuration is relatively high and the performance is relatively stable. The accuracy fluctuates significantly under other weight coefficients, and the performance is not stable. Moreover, the recognition rate of action 5 is not high

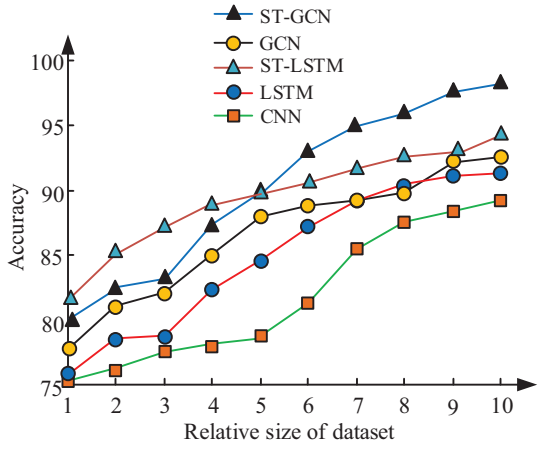
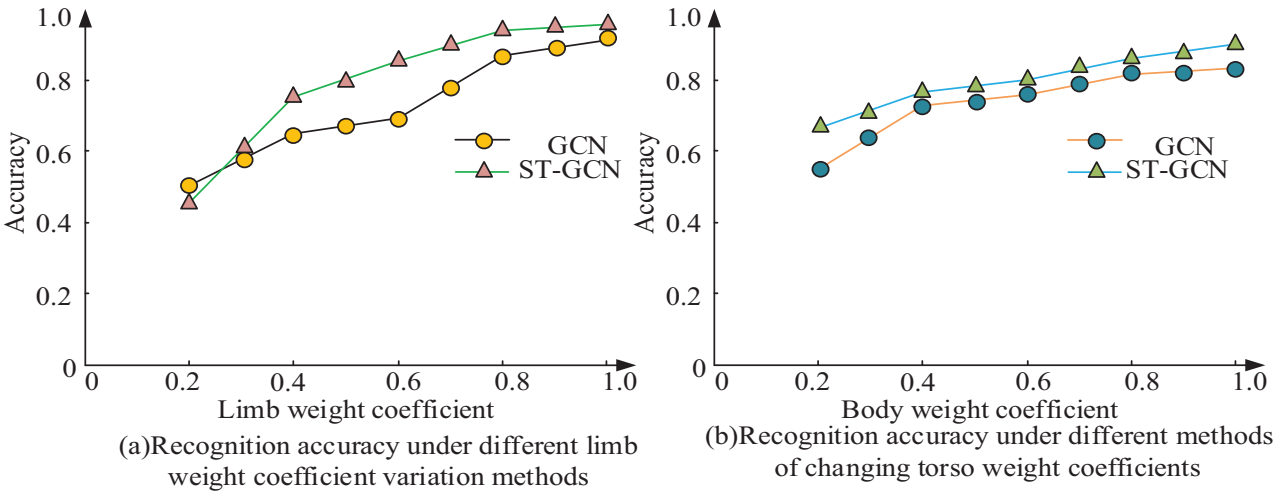


Fig. 9. Accuracy of different methods under different dataset sizes.

for the weight coefficients in each configuration, as the distance between two players in back-to-back singles is usually close, resulting in a significant amount of noise in the action data generated by this action. Considering the accuracy and stability of the model, the third kind of weight coefficient configuration is selected, that is, the trunk weight is set to 0.7, and the limbs weight is 0.8. Introducing different methods to compare the methods used in this study yields (Fig. 12).

In Figure 12, the proposed ST-GCN has a high recognition accuracy for each action, but the accuracy for action 5 is not high. This is because during back-to-back singles, the distance between the two players is relatively close. This, coupled with interference from other players, results in a large amount of noise in the collected data, which seriously interferes with recognition. The average accuracy of CNN, LSTM, GCN, ST-LSTM, and ST-GCN are 61.6%, 65.4%, 72.5%, 76.8%, and 90.3%, respectively. The above data confirms that the proposed ST-GCN method has higher recognition accuracy than other methods and has relatively excellent performance.



(a)Recognition accuracy under different limb weight coefficient variation methods

(b)Recognition accuracy under different methods of changing torso weight coefficients

Fig. 10. Recognition accuracy under different weight coefficients.

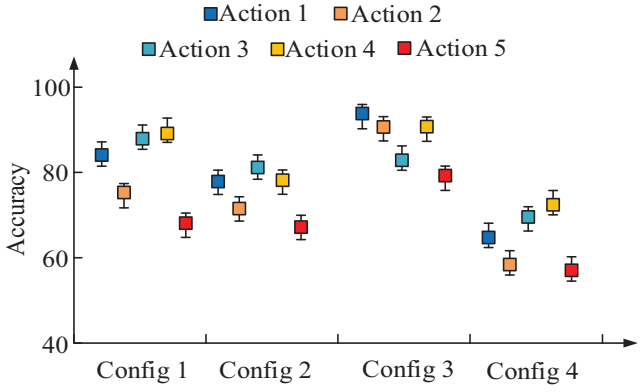
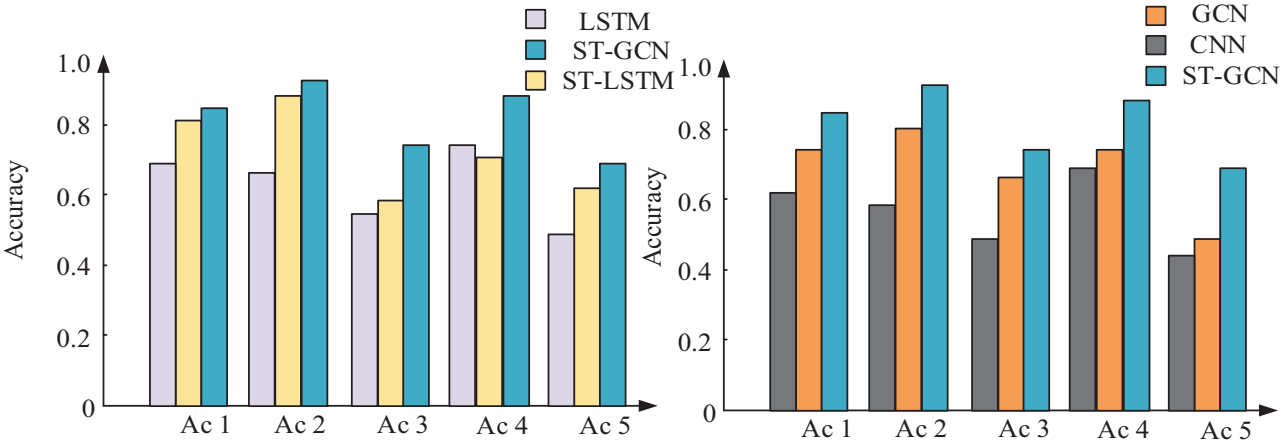


Fig. 11. Accuracy of different basketball movements under different weight coefficients.



(a)Accuracy under different configurations

(b)Accuracy under different configurations

Fig. 12. The recognition accuracy of different algorithms for different basketball movements.

5 Conclusion

In recent years, the rapid development of the economy has led to the rapid growth of the computer industry, and human motion recognition in computer vision has become a hot research direction in this field. This study proposed a combined method of action recognition and 3D pose estimation to recognize and classify basketball movements. The experimental results showed that when the dataset size was 800, the accuracy of the trained ST-GCN model tended to stabilize at 92.5%. When the number of iterations reached 6, the accuracy of theST-GCN model reached 92%. Compared to other methods, the accuracy of this method was 61.6% under the CNN method, 65.4% under the LSTM method, and 72.5% under the GCN method. The accuracy under the ST-LSTM method was 76.8%, and the accuracy under the ST-GCN method was 90.3%. The proposed method showed high performance. The main contributions of this study have two aspects. The first point is to propose a method that integrates HPE with action recognition, providing a new approach for basketball-assisted training. Compared to traditional single action recognition methods, this method improved recognition accuracy and displayed more intuitively. The second point is to study a 3D action recognition method. A multi-person pose estimation method was used to extract 2D skeleton information from basketball basic action video data and convert it into 3D skeleton information. The 3D skeleton data was then input into the STG-CNN model for recognition and classification, resulting in improved accuracy of action recognition. The study still has some limitations, as the recognition effect is not very effective when basketball players move too quickly or are at a distance. If improvements are made in the direction of multi person pose estimation, the performance of the model can be improved.

Conflicts of interest

The author reports there are no competing interests to declare.

Data availability statement

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Author contribution statement

All work for this article was completed by Mingzhi Ye.

References

1. A. Sarkar, A. Banerjee, P.K. Singh, R. Sarkar, 3D human action recognition: through the eyes of researchers, Exp. Syst. Appl. **193**, 1164–1181 (2022)
2. Y. Liu, R. Ma, H. Li, C. Wang, Y. Tao, D. Giovanni, RGB-D human action recognition of deep feature enhancement and fusion using two-stream ConvNet, J. Sens. **202**, 886–895 (2021)
3. Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting, IEEE Trans. Intell. Transp. Syst. **21**, 4883–4894 (2020)
4. X. Yang, Q. Zhu, P. Li, P. Chen, Q. Niu, Fine-grained predicting urban crowd flows with adaptive spatiotemporal graph convolutional network, Neurocomputing **446**, 95–105 (2021)
5. H. Zhou, D. Ren, H. Xia, M. Fan, X. Yang, H. Huang, AST-GNN: an attention-based spatiotemporal graph neural network for interaction-aware pedestrian trajectory prediction, Neurocomputing **445**, 298–308 (2021)
6. X. Zhang, G. Chen, An automatic insect recognition algorithm in complex background based on convolution neural network, Traitement du Signal **37**, 793–798 (2020)
7. Y. Yang, A vehicle recognition algorithm based on deep convolution neural network, Traitement du Signal **37**, 647–653 (2020)
8. X. Song, S. Gao, C. Chen, S. Wang, A novel face recognition algorithm for imbalanced small samples, Traitement du Signal **37**, 425–432 (2020)
9. A. Gharahdaghi, F. Razzazi, A. Amini, A non-linear mapping representing human action recognition under missing modality problem in video data, Measurement **186**, 1101−1109 (2021)
10. B. Sun, D. Kong, S. Wang, L. Wang, B. Yin, Joint transferable dictionary learning and view adaptation for multi-view human action recognition, ACM Trans. Knowl Discov. Data **15**, 32–56 (2021)
11. W. Chen, L. Liu, G. Lin, Y. Chen, J. Wang, Class structure-aware adversarial loss for cross-domain human action recognition, IET Image Process. **15**, 3425–3432 (2021)
12. L. Liu, L. Yang, W. Chen, X. Gao, Dual view 3D human pose estimation without camera parameters for action recognition, IET Image Process. **15**, 3433–3440 (2021)
13. Y. Li, X. Xu, J. Xu, E. Du, Bilayer model for cross-view human action recognition based on transfer learning, J. Electr. Imag. **28**, 1–14 (2019)
14. Z. Tu, H. Li, D. Zhang, J. Dauwel, B. Li, J. Yuan, Action-stage emphasized spatiotemporal VLAD for video action recognition, IEEE Trans. Image Process. **28**, 2799–2812 (2019)
15. W. Xu, M. Wu, J. Zhu, M. Zhao, Multi-scale skeleton adaptive weighted GCN for skeleton-based human action recognition in IoT, Appl. Soft Comput. **104**, 1568–1579 (2021)
16. H.B. Naeem, F. Murtaza, M.H. Yousaf, S. Velastin, T-VLAD: temporal vector of locally aggregated descriptor for multiview human action recognition, Pattern Recogn. Lett. **148**, 22–28 (2021)
17. W. Peng, J. Shi, T. Varanka, G. Zhao, Rethinking the ST-GCNs for 3D Skeleton-based Human Action Recognition. Neurocomputing, 2021, **454** (8): 45–53.
18. F. Li, A. Zhu, Z. Liu, Y. Huo, Y. Xu, G. Hua, Pyramidal graph convolutional network for skeleton-based human action recognition, IEEE Sens. J. **21**, 16183–16191 (2021)
19. X. Ji, Q. Zhao, J. Cheng, C. Ma, Exploiting spatiotemporal representation for 3D human action recognition from depth map sequences, Knowl. Based Syst. **227**, 1057–1069 (2021)
20. Y. Lei, Research on micro video character perception and recognition based on target detection technology, J. Comput. Cogn. Eng. **1**, 83–87 (2022)

Cite this article as: Mingzhi Ye, Application of 3D recognition algorithm based on spatio-temporal graph convolutional network in basketball pose estimation, Int. J. Simul. Multidisci. Des. Optim. **15**, 9 (2024)