Kang D (2024) OPTIMIZATION METHOD OF REAL-TIME BASKETBALL DEFENSIVE STRATEGY BASED ON MOTION TRACKING TECHNOLOGY AND DEEP LEARNING. Revista Internacional de Medicina y Ciencias de la Actividad Física y el Deporte vol. 24 (94) pp. 515-531.
DOI: https://doi.org/10.15366/rimcafd2024.94.032

ORIGINAL

OPTIMIZATION METHOD OF REAL-TIME BASKETBALL DEFENSIVE STRATEGY BASED ON MOTION TRACKING TECHNOLOGY AND DEEP LEARNING

Dandan Kang

Competitive Sports Center, Chongqing College of Mobile Communication, He Chuan District, Chong Qing City, 401520, China
E-mail: 17612313419@163.com

Recibido 28 de abril de 2023 **Received** April 28, 2023
Aceptado 18 de Septiembre de 2023 **Accepted** September 18, 2023

ABSTRACT

Basketball games, real-time adjustment of defense strategy according to the changes on the court can greatly improve the team's chance of success in defense and thus win the game. The development of artificial intelligence technology, especially computer vision, provides a technical basis for real-time tracking of players' positions and body postures. In this paper, a player posture estimation algorithm based on local spatial constraints is proposed based on motion tracking technology and deep learning technology. The algorithm is based on the general human body pose estimation algorithm, which locally constrains the picture based on the athlete detection frame, retains only the content related to the athlete in the picture, and then inputs the processed picture into the general human body pose estimation model for pose detection, and finally maps the detected pose back to the original picture to complete the estimation of the athlete's pose. The model is verified to be able to efficiently complete the real-time defensive strategy optimization task by using NBA game videos for testing.

KEYWORDS: motion tracking; deep learning; pose estimation; defensive strategy

1. INTRODUCTION

Basketball, as a competitive and dynamic sport, puts high demands on players' immediate decision-making and reaction abilities during the game.

Defensive strategies in basketball are crucial for achieving victory, however, traditional defensive strategies often rely on the experience and intuition of coaches and players and lack scientific analysis for real-time data and opponent behavior. In recent years, the rapid development of artificial intelligence and Internet of Things (IoT) technology has brought brand-new technical means to basketball (Chen, Han, Zhang, You, & Zheng, 2023; Chen, Li, et al., 2023; Y. Li & Cao, 2021). Artificial intelligence analyzes basketball game data and can provide coaches with more in-depth data support to improve tactical arrangements and decision making. Meanwhile, IoT technology can also be used to monitor game data in real time to provide real-time feedback and data support for coaches and players (Bilecenoğlu & Yokeş, 2022).

With the development of motion tracking technology and deep learning algorithms, researchers have begun to explore how these advanced technologies can be used to optimize basketball defensive strategies. By analyzing real-time data and motion trajectories on the court and combining them with deep learning models to predict the behavior of opposing players, coaches and players can be provided with smarter and more efficient defensive strategies. The significance of this research is that it is expected to help coaches and players better understand the game situation by providing real-time data support, thus enabling them to make more informed decisions. At the same time, this approach also provides teams with the opportunity to improve their level of competition and tactical execution by utilizing advanced technological tools. In conclusion, the real-time basketball defensive strategy optimization approach based on motion tracking technology and deep learning is expected to revolutionize tactical training and game performance in basketball (Akman, Cairns, Comar, & Hrozencik, 2014).

Computer vision technology, as one of the most rapidly developing directions in the field of artificial intelligence, has been widely applied to basketball. (Mehrasa, Zhong, Tung, Bornn, & Mori, 2018) believe that different players and different teams due to their own sports characteristics and different tactical styles, these characteristics will be hidden in the trajectory of the actual game, can be based on this characteristic, the use of one-dimensional convolutional kernel of the convolutional neural network based on the trajectory characteristics of the players in the game to model, the characteristics of these characteristics will be implicitly extracted, for the identification of the event and the classification of the team. (K.-C. Wang & Zemel, 2016) studied for basketball offensive call tactics, using trajectory sequence data of players' two-dimensional coordinates, using traditional neural networks and recurrent neural networks in machine learning to train modeling of these planar coordinate trajectories for classification of basketball tactics used for dynamic analysis, with good classification results for the season of training as well as for seasons different from the one used for training. (Ramanathan et al., 2016) found that only a small fraction of multiplayer event recognition in fact works for events.

Accordingly, they used a recurrent neural network, combined with a time-varying attention mechanism with learnable weights to construct a new model that can learn to detect events in a video while automatically focusing on the main person of the event, and experimentally demonstrated the effectiveness of their model. (Felsen & Lucey, 2017) proposed a new attribute-based representation of a basketball player's body posture while shooting a three-pointer to capture the player's movements during different phases of the shot. They used Pearson's Chi-squared test to standardize the distribution of attributes for shooting hits and misses and analyzed in-depth the shots of shooting three-pointers. Statistically significant differences were found between the distributions of attributes describing the style of movement (e.g., walking, running, or jumping) and those describing the quality of the delivery, direction of movement, and footwork during different phases of the pitch and in different game situations.

However, due to the characteristics of basketball itself, computer vision technology also faces numerous challenges. First of all, the environment of computer vision technology basketball game is complex and variable, including factors such as player interactions, rapid movement, and occlusion. For deep learning algorithms, many difficulties need to be overcome to deal with this complex environment and provide accurate defense strategies. Second, in basketball games, defensive strategies need to be adjusted in real-time changing situations. Therefore, it is challenging to ensure that deep learning-based systems can quickly and accurately analyze tracking data and generate real-time recommendations. Finally, obtaining high-quality basketball game data and processing it effectively is one of the challenges. This may involve multi-camera tracking of information such as player positions, movements, etc. and transforming it into data suitable for deep learning model training. The purpose of this paper is to introduce a real-time basketball defensive strategy optimization method based on motion tracking technology and deep learning, which analyzes the real-time data and motion trajectory on the court and combines with the deep learning model to predict the behaviors of the opponent players, so as to propose a more intelligent and efficient defensive strategy. The method cannot only help coaches and players better understand the game situation, but also provide them with references to make smarter decisions during the game, which is expected to play an important role in improving the team's competitive level and tactical execution ability.

2. Effectiveness and Problems of Top-Down Player Posture Estimation

The top-down multi-person pose estimation algorithm first performs target detection of the human body in the image, then crops the image based on the human body detection frame, then performs pose estimation using a single-person pose detector, and finally maps the poses of all people back to the original image. This method decomposes multi-person pose detection into multiple single-person pose detection based on the detection frame and single-person pose detector, which can achieve good results in the case of dispersed people. The athlete-based detection frame adopts the top-down idea for athlete's pose detection, the athlete's pose can be obtained directly, and the additional player differentiation module is no longer needed.

2.1 Algorithmic framework

The top-down human pose estimation idea is combined with the previous detection of athletes, and its complete detection flow is shown in Figure 1. Firstly, the video is sub-framed, and then the person in each frame is detected using a generic target detector based on Faster-RCNN. Then the trained example feature metric model is used to distinguish the detection results of athletes. Then the detection results of each frame are optimized using a multi-frame detection result fusion strategy based on the video time domain context. Then the image is cropped with some degree of zoom based on the athlete's detection frames and fed to the single person pose detector. Finally, the results detected by the single person pose detector module are mapped back to the original image to obtain the complete athlete's human pose estimation for each frame.

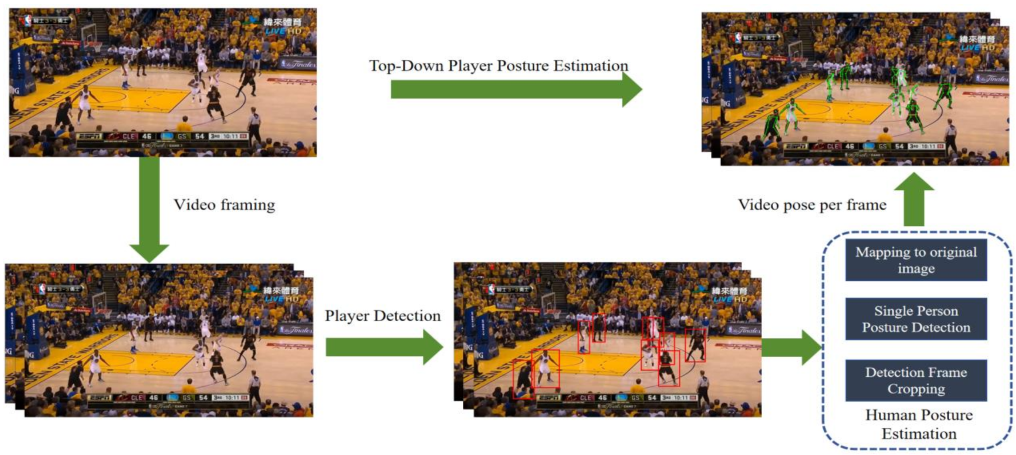

Figure 1: Top-down framework of player pose estimation

2.2 Selection and Effectiveness of Single Person Posture Detectors

The performance of the top-down multiplayer pose estimation algorithm is mainly determined by two factors: the performance of the target detector and the performance of the single-person pose detector. The omission and misdetection of the target detector will directly lead to the omission and misdetection of the human posture, while the accuracy of the target detector localization also affects the detection results of the single-person posture detector. In this paper, Faster-RCNN+MIL+Merge player detection algorithm and Alphapose multi-person pose estimation algorithm are used for top-down player pose estimation, and the results are shown in Figure 2. From the figure, it can be seen that the top-down pose estimation based on the athlete's detection frame can effectively recognize the athlete's pose, and it can also be seen that the results of athlete's pose estimation are more ideal when the people are more dispersed, but there will be obvious misdetections when the people are gathered.

(a) Effectiveness of player dispersion detection

(b) Effectiveness of centralized player testing

Figure 2: Results of top-down player pose estimation

2.3 Limitations of the top-down approach

Based on the target detection results of the athletes in the sports video, the top-down human posture detection algorithm can be used to accomplish the estimation of the human posture of the athletes in the sports video, but this method has the following two problems: 1) A single person pose estimation is performed for each target detection result, and the time to perform the pose estimation increases linearly as the number of detections in the picture increases. (2) As shown in Figure 3, when there is a gathering of people, because the detection frame performs NMS (Non-Maximum Suppression), there will be a situation where multiple targets are in one detection frame. In this case, the single person pose estimation based on the detection frame can only get one human body pose, which will lead to missed or false detection.

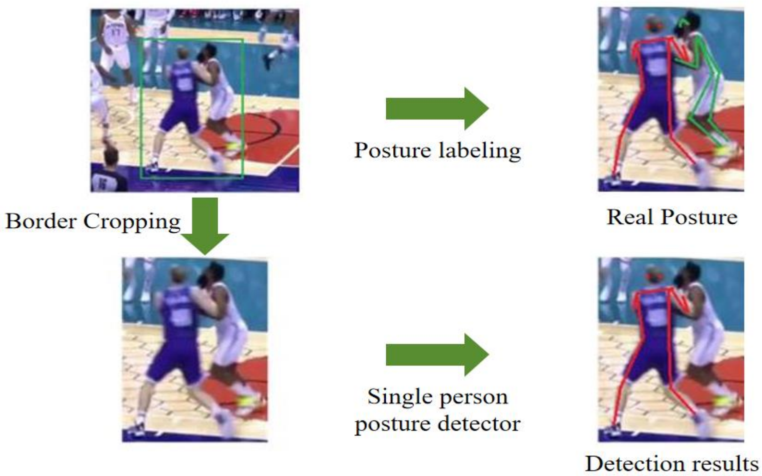

Figure 3: Multiple targets in one box

3. Effectiveness and problems of bottom-up player posture estimation

Situations such as multiple people gathering and occlusion appear more often in basketball games. In order to solve the above-mentioned problems, the idea of bottom-up pose estimation is proposed, i.e., regressing the joints of all the people in the graph at one time, and then completing the matching connection of the joints to get the human body poses according to the strategies of nearest-neighbor matching and greedy search, etc. This method does not rely on the detection frame, which is more suitable for the situation of dense human body.

3.1 Algorithmic framework

The bottom-up player pose estimation framework is shown in Figure 4. The algorithm first detects the joints of all the people in the diagram, and then performs joint matching to compose the joints into different human body postures, and then performs player posture matching based on the detection frame of the player to remove the human body postures that do not belong to the athlete, and finally obtains the posture estimation of the player in the diagram. First return to the joints of the whole figure, and then assemble different human body postures from the joints, this bottom-up idea no longer relies on the detection frame, which can solve the problem of missed detection caused by the gathering of people, and also no longer need to call the single person posture detector several times.

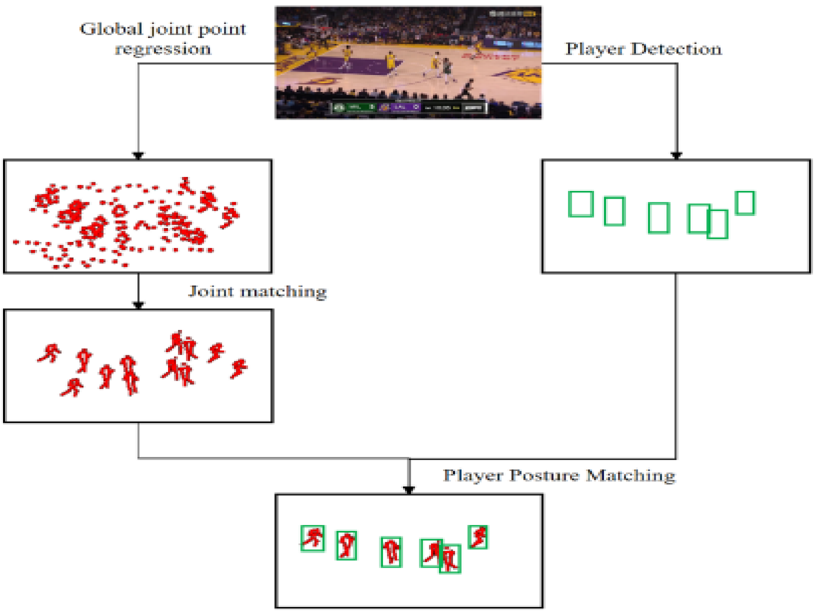

Figure 4: Bottom-up framework of player pose Estimation

3.2 Joint Matching

Bottom-up multi-person pose estimation algorithms can theoretically well reduce the problem of human pose leakage in multi-person aggregation scenarios, but they also bring the problem of human joint point matching, i.e., which human body the joint points should belong to, and how to connect them into different human poses. Traditional algorithms usually construct a probabilistic graph model based on the prior distribution of human joints (J. Wang et al., 2008), and then calculate the connection strengths between different joints based on the probabilistic graph for joint connection, but this method relies on the distribution of the dataset and cannot be effectively generalized. In the existing joint matching algorithms, the most common practice is to regress the connection strengths between joints when regressing the joints, and then conduct joint matching based on the connection strengths between different joints to obtain different human body postures.

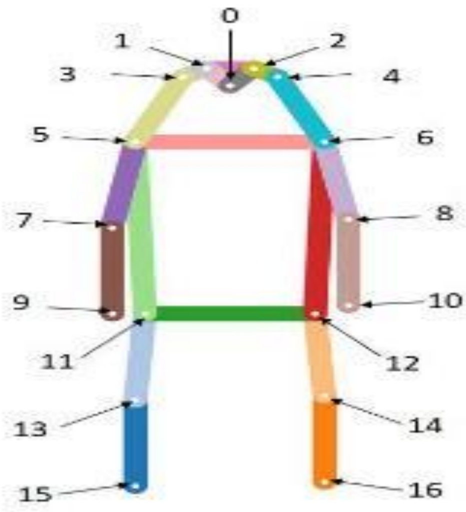

Figure 5: Open Pifpaf skeleton model

This paper is based on the Pifpaf (Kreiss, Bertoni, & Alahi, 2019) multiplayer pose estimation algorithm to accomplish bottom-up player pose estimation, which is modeled in the Openpifpaf skeleton framework in Figure 5. The algorithm regresses both PIF (Part Intensity Field) for human joint point localization and PAF (Part Association Field) for human joint point connection. The coordinates and confidence levels of 17 types of joints in the figure can be obtained from the PIF, and 19 types of joint connection strengths can be obtained from the PAF. The joint matching strategy based on joint coordinates, confidence level and connection strength is as follows.

1) Noise removal: Sort the joints based on the confidence of the joints from the largest to the smallest, and then set a threshold to remove the joints whose confidence is lower than the threshold.

2) Posture Generation: The gesture generation strategy used in this paper is a greedy algorithm, i.e., each joint point is treated as a seed for gesture generation, and the gesture generation rules are similar to the minimum

spanning tree of a graph. A pose consists of 17 different classes of joint points, take one of the classes of joint points A as an example, its generated pose is a collection of joint points of multiple classes, with at most one joint point of each class. At the beginning, the generated pose set of this joint point contains only joint point A. Calculate the connection strengths of the joint points in the set and the joint points of all the categories not contained in the set, find out the joint point with the largest connection strength, and when the connection strength is larger than a set threshold, add it to the generated pose set of A, otherwise, do not add it. According to the above principle, new categories of joints are added to the pose set of joint A until all 17 categories of joints are added or the connection strength of all remaining joints is less than the set threshold. Because the number of human bodies in the graph is uncertain, and the number of regression joints is also uncertain, it is not guaranteed that the generated poses are all complete, and some poses may only contain some human body joints.

3) Gesture de-duplication: The gesture generation based on greedy algorithm will cause a large number of gesture duplications, so the duplicated gestures should be removed according to certain rules, and the idea of de-weighting is consistent with the target detection frame NMS. A gesture can be regarded as a collection of multiple joints, similar to the target detection frame IoU, the IoU between the gestures can be regarded as the intersection and ratio of the two collections, and at the same time take the average of the confidence level of the joints in the collection as the confidence level of the gesture, then the gesture can be subjected to NMS.

3.3 Attitude Matching and Detection Effect

The top-down player pose estimation algorithm, based on the athlete's detection frame, so the pose obtained by the single person pose detector is the athlete's pose, but the bottom-up multi-person pose estimation algorithm, which does not rely on the human detection frame, outputs poses that are the poses of all the human bodies in the graph, and therefore also distinguishes between athletes and non-athletes for these poses.

The detection frame of the player represents the positional distribution of the athlete's human body. The human body pose can be viewed as a collection of points that represent the positions of the joint points of the human body, and also the outer rectangles made up of these points represent the positions of this human body. Because the joints are not always at the edges of the body, the body position represented by these outer rectangles will be smaller than the detection frame for target detection. Therefore, when a human body gesture falls within the detection frame of the athlete, the gesture can be recognized as belonging to the athlete. All the poses are matched one by one with the athlete's detection frame, and when there exists a detection frame that completely contains the pose, the pose is considered to belong to the athlete's pose for retention, otherwise the pose is removed. The results after matching the postures based on the player detection frame are shown in Figure 6, where it can be seen that the bottom-up player posture estimation algorithm detects well even in the case of people gathering.

Figure 6: Results of bottom-up player pose estimation

3.4 Limitations of the bottom-up approach

Compared with the top-down approach, the bottom-up approach solves the problem of repeated calls to single detectors and the problem of missed detections caused by the aggregation of people, but it also increases the process of matching athletes' postures.

In the player pose estimation task, only the athlete's pose information is needed, but the bottom-up approach computes the poses of all people in the graph. In the basketball video dataset constructed in this paper, the maximum value of the number of people to be detected in each picture is 10, but the number of people contained in the picture is much larger than this value, which means that the bottom-up player pose estimation method performs a large number of unnecessary operations, resulting in a waste of arithmetic power and time overhead.

4. player pose estimation under local spatial constraints

Aiming at the problems of top-down player pose estimation algorithm missing detection in the case of personnel gathering and bottom-up player pose estimation algorithm increasing additional computational overhead, this paper proposes a player pose estimation algorithm based on local space constraints. The algorithm is based on the bottom-up pose estimation algorithm, and the athlete's pose can be directly output through certain preprocessing, which reduces the matching process and accelerates the detection speed while ensuring the detection accuracy.

4.1 Algorithmic framework

The bottom-up pose estimation algorithm does not rely on the human body detection frame, which by default detects the poses of all the people in the graph because only the pose estimation of the athletes is needed, which generates a large number of invalid computations and brings about the problem of matching the poses of the athletes. The flow of the player pose estimation algorithm based on local space constraints proposed in this paper is shown in Figure 7. The method is based on the detection frame of the player to carry out certain preprocessing of the input image, so that the effective content of the input image contains only the athlete, and then input the image into the deep network for joint regression and matching, and finally directly get the player pose in the figure. This method does not need to perform player pose matching, and greatly reduces the number of joints in the whole figure, reducing the computational overhead of joint matching.

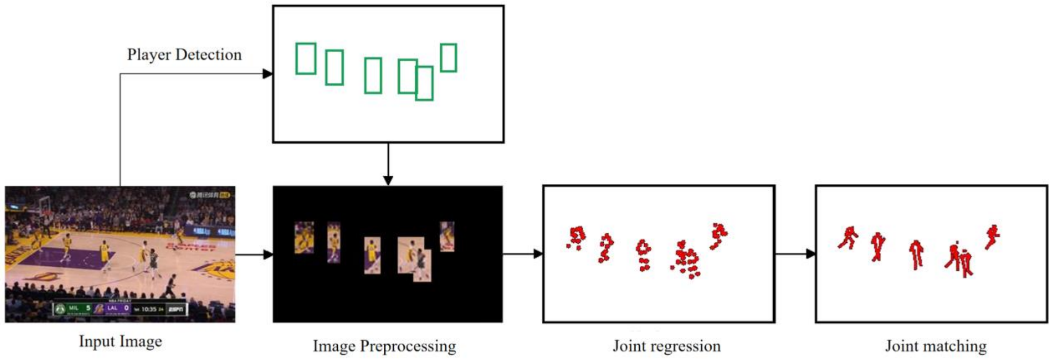

Figure 8: Pose estimation of player based on local space constraints

4.2 Theoretical analysis and image preprocessing

As can be seen in Figure 8, the image preprocessing is based on the image of the player detection frame to remove the background that does not contain the player (the background part of the image pixels will be set to 0), and the image that only contains the player content as the input, into the deep network for detection. If there is only a player in the image, then the detection result is naturally the athlete's pose, and the computation amount of joint matching is greatly reduced and player pose matching is no longer needed. However, it is important to consider whether the content of the input image will have any effect on the detection result after the content is changed.

In the field of deep learning, generally images are subjected to multiple convolutional layers for feature extraction, which are then employed in tasks such as image classification and target detection. In the image classification task, the focus is on the category of the whole image, and all the contents in the image may have an impact on the classification result. In the target detection task, the focus is on the detected object in the image, and the goal is to recognize the object in different scenarios, when different backgrounds have little effect on the detection results. In the human joint point regression task, the focus is on the joint points of the human body. In terms of content, the judgment of whether the point is a human joint is mainly based on the image content of the human body part, and other parts of the image content has little impact. In terms of computation, after multiple convolutional layers of feature extraction, the feature receptive field (X. Li, Wang, Hu, & Yang, 2019) is Gaussian distributed, and the contribution of the image content is inversely proportional to the distance to the point, i.e., the image content that is far away from the location of the joint point has little effect on the detection results. In summary, theoretically, preprocessing the input image to remove the background other than the athlete's body has little effect on its player pose estimation results. This paper is based on the player detection frame and Pifpaf multi-person pose estimation algorithm for player pose estimation under local space constraints, and its detection effect is shown in Figure 9, from which it can be seen that the method can successfully detect the human body pose, and the athlete's pose can be directly obtained without computations such as pose matching.

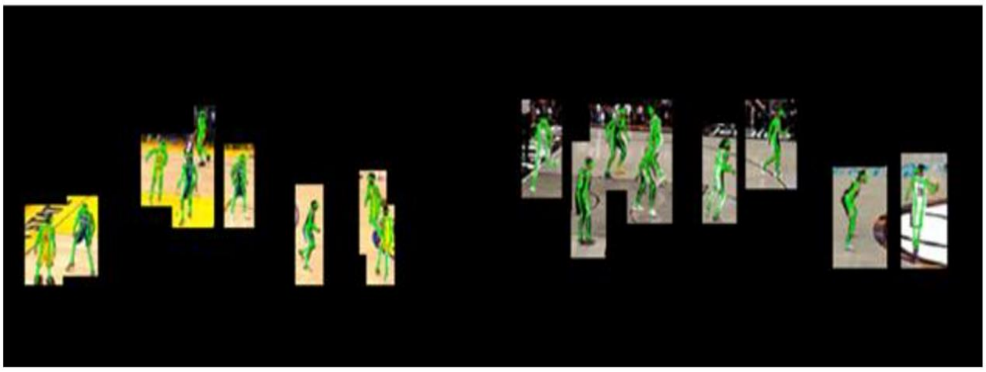

Figure 9: Results of pose estimation without background

4.3 Algorithm complexity analysis

player pose estimation based on local spatial constraints is based on the bottom-up pose estimation algorithm, and the two algorithms are consistent in the computation of joint point regression and have the same algorithmic complexity. Therefore, we mainly analyze the algorithm complexity of the two algorithms in the three parts of joint point matching, player pose matching and image preprocessing. In order to facilitate the analysis, this paper does not compare the specific computational details in depth, and mainly compares the part of which requires cyclic computation.

Assuming that the total number of human bodies contained in an image is N , and the number of athletes is M, then the number of loops required for image preprocessing is $N \times M$, the number of loops for player pose matching is, and the number of loops for joint point matching is $N \times 17 \times M \times 17$. Thus, the total number of loops of the top-down player pose estimation algorithm is $N \times M + N \times 17 \times M \times 17$, and the total number of loops of the player pose estimation algorithm based on local space constraints is $M + N \times 17 \times M \times 17$. Take the NBA basketball video dataset constructed in this paper as an example, take the maximum value of the number of athletes in each picture, $M = 10$, and the mean value of the total number of human bodies in each picture is $N = 40$. The complexity of the top-down player pose estimation algorithm is 462,800, and the complexity of the player pose estimation algorithm based on local constraints is 28,910, which shows that the player pose estimation algorithm based on local constraints can effectively reduce the amount of computation to reduce the complexity of the algorithm. The experimental part of this paper will compare the specific time spent on each part.

5. Experimental results and analysis

5.1 Experimental environment

In order to conduct a reasonable and fair test, all algorithms in this experiment are performed in the same hardware and software environment. Running environment: The CPU model used in this experiment is Intel Core i7-8750, the GPU model is GTX1060 (6G), the operating system is Windows10, the programming language is Python3.6.2, the development tool is Pycharm2018, and the deep learning framework is PyTorch0.4.0.

5.2 Evaluation indicators

There are two ways of evaluating human joint points, PCK (Andriluka, Pishchulin, Gehler, & Schiele, 2014) (Percentage of Correct Keypoints) and OKS (Object Keypoint Similarity). In this paper, we choose the more commonly used OKS method, which calculates the similarity between two poses and gives a score between 0-1. In this paper, the OKS score is calculated between the predicted pose and the real pose, and when the OKS>0.5, the pose of the player is considered to be correctly detected.

player detection and stance estimation are both a kind of detection tasks, this paper selects three common evaluation indexes based on the evaluation of detection tasks, which are maximum recall, AP value and P-R curve, which are defined as follows. For a detection result, the predicted value and the true value may have four cases: TP, FP, FN, TN, TP means that the positive sample is predicted to be true, FP means that the negative sample is predicted to be true, FN means that the positive sample is predicted to be false, and TN means that the negative sample is predicted to be false. In the P-R curve, P means Precision and R means Recall, and the formula is as follows:

$$P = \frac{TP}{TP+FP} \quad (2\text{-}2)$$

$$R = \frac{TP}{TP+FN} \quad (2\text{-}3)$$

5.3 Algorithm speed comparison experiment

In this experiment, the player pose detection algorithm proposed in this paper is subjected to a unified speed test on the NBA basketball video dataset. 100 pictures are tested, with an average number of 7.73 athletes per picture. There are 100 pictures in the test, and the average number of athletes per picture is 7.73. In order to conduct a reasonable and fair test, all algorithms in this experiment are carried out under the same hardware and software environment. Because different algorithms are different in part of the data processing, this experiment will test the complete detection time of each algorithm and the running time of each main module alone In order to ensure the stability of the experimental results, after all the pictures in the test set are detected, the average value of each time is taken as the test result.

The algorithms compared in this experiment are 1) a top-down player pose estimation model based on Alphapose, 2) a bottom-up player pose estimation model based on Pifpaf, and 3) a player pose estimation model Local Pifpaf based on Pifpaf introducing local spatial constraints. In this experiment, in addition to comparing the average single-image complete detection time of each algorithm, we also compare the detection time of each main module, which are image preprocessing, deep network computation, joint matching and player pose matching, and the speed comparison results of each algorithm are shown in Table 4-1.

Table 4-1: Speedof different poseestimation methods（ms）

MODEL	IMAGE PREPROCESSIN G	DEEP NETWORK COMPUTING	JOINT MATCHI NG	PLAYER POSTURE MATCHING	DETECTI ON TIME
ALPHAPOS E	0	156	-	-	205
Pifpaf	0	259	105	10	447
LocalPifpaf	0	208	32-	-	302

From Table 4-1, it can be seen that the main time spent on athletes' posture estimation is spent on neural network computation, in which the time spent on some modules is 0. This is because the smallest unit of timing is milliseconds (ms), and no valid comparison can be formed when the time spent is less than 1ms, so the time spent on this part of the time is ignored and uniformly set to 0.

The fastest detection speed is the top-down pose estimation algorithm, although the detection time of this algorithm increases linearly with the number of people, but because of the fast speed of its single-person pose detector, it still maintains a certain speed advantage when the average number of people in the detection picture is about 7.73. The local spatial constraint-based pose estimation algorithm and the bottom-up pose estimation algorithm do not spend much time difference in the neural network computation part, but in the image preprocessing, joint matching and player pose matching parts, it is much faster than the bottom-up pose estimation algorithm, and it can also improve the speed of more than 30% in the complete detection time. This indicates that the local space constraint-based pose estimation algorithm effectively reduces the algorithm complexity after preprocessing the image.

5.4 Algorithm performance comparison experiment

In order to test the effectiveness of the method in this paper, the following three player pose estimation algorithms were experimented based on the constructed NBA basketball video dataset:

1) Alphapose-based top-down player pose estimation algorithm; 2) The bottom-up player pose estimation algorithm based on Pifpaf; 3) Local Pifpaf, a player pose estimation algorithm that introduces the local spatial constraints proposed in this paper. This paper calculates the AP values of these three algorithms at OKS=0.5, as shown in Table 4-2, and draws the P-R curves of these three methods at this time to visualize the performance difference of the three methods, as shown in Figure 10.

Table 4-2 Pose estimation results of different methods

MODEL	BAKCBONE	MSCOCO (%)	AP (%)
Alphapose	Hourglass-104	73	79.1
Pifpaf	ResNet-50	64	79.8
LocalPifpaf	ResNet-50	-	80.3

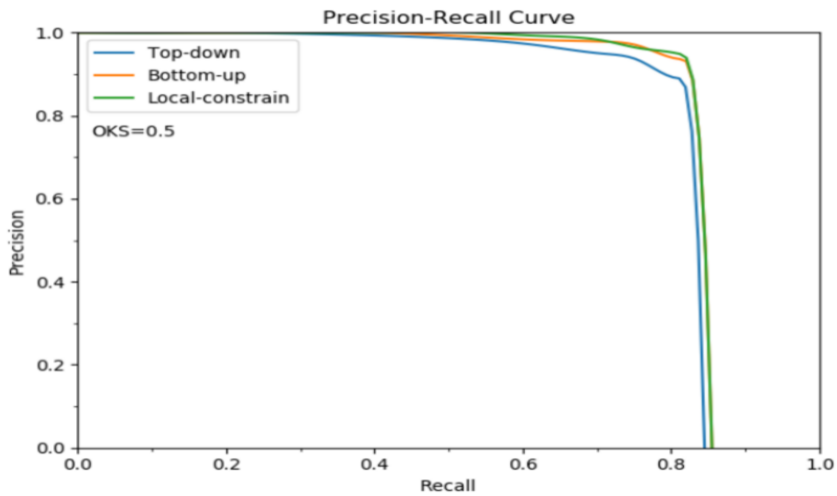

Figure 10: The P-R curve of the pose estimation results

Combining the data in Table 4-2 and Table 4-1, it can be seen that the three players pose estimation algorithms are slightly lower in performance than player detection, which indicates that the correct rate of estimation of athlete's pose is relatively high when the target detection is correct. Compared with the top-down player pose estimation algorithm the bottom-up player pose estimation algorithm has a slight improvement in performance, which indicates that the bottom-up detection idea improves the performance of pose detection in the case of people gathering and occlusion in sports videos.

The player pose estimation algorithm based on local spatial constraints maintains a comparable performance to the bottom-up player pose estimation algorithm while improving the detection speed, which indicates that the non-player human body background removed in the image preprocessing has little effect on the athlete's pose detection, and proves the validity of the method. As can be seen from the P-R curves in Figure 10, the bottom-up multi-person pose estimation algorithm has a better performance in the field of sports videos where there are more people gathering and more occlusion situations.

6. Simulate basketball defense strategy optimization

By inputting the NBA game video into the player posture estimation model Local Pifpaf proposed in this paper based on Pifpaf introducing local spatial constraints, we can clearly analyze the conversion process of both sides' defensive strategies during the game, as can be seen in Figure 11. Through the cooperation between the athletes, how the attacker mobilizes the movement of the defender through the mutual cover of teammates and running without the ball, so that the defender's defensive center of gravity is shifted and loopholes appear, and then find the opportunity to attack. How the defense can delay the opponent through the three forms of cooperation, such as bypassing, stealing and switching; how the defense can defend the player without the ball through the three forms of bypassing, switching and crossing; and how to use the staunch defense to defend the position, etc. can be shown more quickly. At the same time, each player's trajectory, speed, action and other data can be recorded in real time, so as to analyze the physical exhaustion of the athletes, and reasonably arrange the substitution tactics.

Figure 11: Sample Schematic of NBA Game Video Analysis

7. Conclusion

This paper proposes a Local Pifpaf model based on deep learning technology to optimize the application of motion tracking and the implementation of defensive strategy in basketball.

The model can effectively track the players on the court and obtain real-time body posture information, which can efficiently provide coaches with the factual state information of the players on the court and assist coaches in making timely adjustments to the defense strategy. At the same time, it can provide coaches with information on the physical exertion level or injury situation of the players, so as to avoid serious injuries to the players.

Reference

Akman, O., Cairns, D., Comar, T. D., & Hrozencik, D. (2014). Integrated pest management with a mixed birth rate for prey species. *Letters in Biomathematics, 1*(1), 87-95. doi:https://doi.org/10.30707/LiB1.1Akman

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). *2d human pose estimation: New benchmark and state of the art analysis.* Paper presented at the Proceedings of the IEEE Conference on computer Vision and Pattern Recognition.

Bilecenoğlu, M., & Yokeş, M. (2022). Distribution of two closely allied gobies, Gobius bucchichi Steindachner, 1870 and Gobius incognitus Kovačić & Šanda, 2016, along the Turkish coast. *FishTaxa, 23*, 47-53.

Chen, J., Han, P., Zhang, Y., You, T., & Zheng, P. (2023). Scheduling energy consumption-constrained workflows in heterogeneous multi-processor embedded systems. *Journal of Systems Architecture, 142*, 102938.

Chen, J., Li, T., Zhang, Y., You, T., Lu, Y., Tiwari, P., & Kumar, N. (2023). Global-and-local attention-based reinforcement learning for cooperative behaviour control of multiple UAVs. *IEEE Transactions on Vehicular Technology*.

Felsen, P., & Lucey, P. (2017). *Body shots: Analyzing shooting styles in the nba using body pose.* Paper presented at the Proceedings of the MIT sloan sports analytics conference, possession sketches: mapping NBA strategies, Boston, MA, USA.

Kreiss, S., Bertoni, L., & Alahi, A. (2019). *Pifpaf: Composite fields for human pose estimation.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Li, X., Wang, W., Hu, X., & Yang, J. (2019). *Selective kernel networks.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Li, Y., & Cao, J. (2021). WSN node optimal deployment algorithm based on adaptive binary particle swarm optimization. *ASP Transactions on Internet of Things, 1*(1), 1-8.

Mehrasa, N., Zhong, Y., Tung, F., Bornn, L., & Mori, G. (2018). *Deep learning of player trajectory representations for team activity analysis.* Paper presented at the 11th mit sloan sports analytics conference.

Ramanathan, V., Huang, J., Abu-El-Haija, S., Gorban, A., Murphy, K., & Fei-Fei, L. (2016). *Detecting events and key actors in multi-person videos.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Wang, J., Fleet, D., Hertzmann, A., Urtasun, R., Geiger, A., Popovic, J., . . . Fua, P. (2008). Gaussian process models for human pose and motion. *IEEE transactions on pattern analysis and machine intelligence, 30*(2), 283-298.

Wang, K.-C., & Zemel, R. (2016). *Classifying NBA offensive plays using neural networks.* Paper presented at the Proceedings of MIT Sloan sports analytics conference.