Hindawi
Scientific Programming
Volume 2021, Article ID 4102294, 11 pages
https://doi.org/10.1155/2021/4102294

Research Article

Football Player Posture Detection Method Combining Foreground Detection and Neural Networks

Xin Hu

School of Football, Xi'an Physical Education University, Xi'an 710068, Shaanxi, China

Correspondence should be addressed to Xin Hu; huxinsport@163.com

Received 29 April 2021; Revised 30 May 2021; Accepted 5 June 2021; Published 22 June 2021

Academic Editor: Shah Nazir

In recent years, with the rapid development of artificial intelligence, information technology, intelligent digital video surveillance systems, real-time sports competition playback, and other technologies have emerged one after another, making the advantages of deep learning-based football posture detection tasks become more obvious. Related models and methods have been applied to the research field of sports posture estimation and have achieved great improvement, surpassing the traditional football posture estimation method based on manual design features in one fell swoop. In addition, the application of video foreground detection has developed rapidly and has great application value in sports analysis. Therefore, this paper proposes a novel football motion detection approach combining foreground detection and deep learning for real-time detection of football player posture. The main task of foreground target detection is to extract the interesting foreground target in the real monitoring scene and use it as the target of interest for subsequent analysis. Then, we propose a triple DetectNet detection framework based on deep learning technology, which can quickly and robustly realize the three-dimensional pose estimation of multiperson motion. For input, the triple DetectNet framework uses three neural networks and is executed in three stages; the first stage is to use the DetectNet (DN) network to detect the bounding box of each person separately, the second stage uses the 2DPoseNet (2DPN) network to estimate each of the corresponding two-dimensional poses of the individual, and the third stage uses the 3DPoseNet (3DPN) network to obtain the 3D pose of the person. This paper also conducted experiments on four datasets, and the results proved the superiority and success of this algorithm.

1. Introduction

In recent years, with the rapid development of artificial intelligence technology, the emergence of intelligent digital video surveillance systems, real-time sports competition playback, and other technologies have made the advantages of computer vision tasks based on deep learning more and more obvious, and it has high flexibility and openness, which indicates the development direction of intelligent image processing. With the development of deep learning, related models and methods have been applied to the field of football player posture estimation and have achieved great improvement, surpassing traditional posture estimation methods based on hand-designed features in one fell swoop. The football player posture estimation based on the deep learning method has made breakthroughs in various aspects. The problem of moving target detection and tracking based on deep learning has developed in the development of science and technology and engineering applications and has a certain research and application foundation in the fields of football competition, football training, and artificial intelligence research. As far as video playback and monitoring are concerned, if we use deep learning-based related motion detection and tracking technology, we can perfectly assist the competitive team or coach team in completing the task. Since 2009, Hinton et al. [1] published important research work on deep belief networks; deep learning [2–7] has become a new direction of machine learning, and it has been used in the handling of many problems in the field of artificial intelligence. So far, deep learning-related frameworks have developed into a variety of targeted methods, such as convolutional neural networks, deep neural networks,

recurrent neural networks, and deep belief networks, which have been successfully applied in various fields of computer and obtained excellent results. As a result, artificial intelligence has been fully developed.

Football player posture estimation is a basic problem in computer vision. It is the basis of multiperson posture estimation, behavior recognition, and sports goal analysis. It can be widely used in many meaningful fields, such as sports competition and behavior analysis. The goal of soccer player single pose estimation is to find the coordinates of each joint point of the player from the image or video containing a single soccer player. Affected by the shooting angle, scene, lighting, wearing, etc., the estimation of the single athlete's posture in the image is facing arduous challenges. However, with the opening of pose estimation data sets and the development of computer hardware and deep learning technology [8–12], various networks such as convolutional neural networks based on deep learning have gradually penetrated into various research fields of computer vision, and single athlete pose estimation has also been achieved. Since the methods used for pose estimation are mainly based on deep convolutional neural networks, a small number of methods are based on generative adversarial networks. At the same time, based on single-person pose estimation, we have developed multiperson pose estimation, which is accurate from the input RGB video, detects the target person, predicts the 2D key points, and finally predicts the accurate 3D key points through our network to get the final multiperson pose. Based on the existing hardware, we can fully realize real-time multimotion target analysis.

In addition, the application of video foreground detection is also very extensive. The feature is that in competitive sports, its complex dynamic scenes, color camouflage, illumination changes, and static foreground have brought various difficulties and problems to video foreground detection. Foreground target detection algorithms are mainly divided into three categories: detection algorithms based on target modeling, foreground detection algorithms based on background modeling, and detection algorithms combined with deep learning. Arghavan et al. [13] optimized, improved, and, based on the motion information, extracted the feature points corresponding to the moving target from the next frame. Then, the number of moving objects in each frame is determined according to the motion information and location, and then the k-means algorithm is used for clustering. The moving target is clustered using the feature vector composed of pixel intensity, motion amplitude, motion direction, and feature point position. This algorithm has high accuracy in determining the number of moving targets, but it cannot detect contour information well. Therefore, detection algorithms based on target modeling are more inclined to detect speed and can detect the position of moving targets, but they cannot effectively detect target contours and easily lose information. The basic principle of the detection algorithm based on background modeling is to compare the current frame image information with the established background model and extract the difference area as the foreground target. But with the development of artificial intelligence, it is a new research idea to integrate deep learning ideas with foreground detection algorithms.

Based on the above observations, this paper proposes a novel soccer motion detection method combining foreground detection and deep learning for real-time detection of football players' posture. The main task of foreground target detection is to extract the interesting foreground targets in the real monitoring scene and use them as interesting targets for subsequent analysis. Then, we proposed a triple DetectNet detection framework based on deep learning technology, which can quickly and robustly realize the three-dimensional pose estimation of multiperson motion. For input, the triple DetectNet framework uses three neural networks and is executed in three stages: the first stage is to use the DetectNet (DN) network to detect the bounding box of each person separately; the second stage uses the 2DPoseNet (2DPN) network to estimate each the corresponding two-dimensional pose of the individual and the third stage uses the 3DPoseNet (3DPN) network to obtain people's 3D pose. Following are the main contributions points of this paper:

(i) To propose a foreground detection method based on confidence weighted fusion and visual attention, which is used to solve the problem of color camouflage and static foreground in the background subtraction method.

(ii) To use target design based on DetectNet detection selects the corresponding 2DPoseNet (2DPN) network framework for two-dimensional pose prediction.

(iii) To present a novel triple DetectNet detection framework to achieve the prediction from two-dimensional pose to three-dimensional pose and obtain better results than existing advanced methods in training.

(iv) To conduct experiments on three data sets, and the results proved the superiority of this algorithm.

The paper is structured as follows: section 2 represents the related work to the proposed research. The research methodology of the proposed study is given in Section 3, with details of the approach used. The experiments and results are given in Section 4. The conclusion of the paper is presented in Section 5.

2. Related Work

Target detection and tracking based on deep learning: first is the problem of target detection, that is, determining the position of the target object in the image or scene, which is generally determined by the bounding box of the object. In response to such problems, Sande et al. [14] proposed that RCNN uses a selection domain method to obtain local candidate regions that may have detection targets in the image, and then input these candidate regions into the convolutional neural network to obtain their features and connect the classifiers Go to the feature map to determine

whether the corresponding area belongs to the target to be detected, and finally perform regression on the calibration frame to correct the position of the prediction frame, but RCNN has the problem of repeated calculation. He et al. [15] introduced the spatial pyramid pooling layer into CNN and proposed SPPnet, which reduced the CNN network's limitation on the input image size and improved the accuracy. Based on the idea of SPPnet, Girshick et al. [16] also proposed a one-stage method Fast-RCNN, based on an adaptive pooling method, mapping the final candidate region to the feature map of the last convolutional layer of the convolutional network. The benefits are obvious. Only one feature extraction is required to perform the detection task, which greatly improves the detection speed. However, Fast-RCNN also has this defect; that is, it takes a long time to extract feature candidate frames; Zhou et al. [17] propose a new detection method whose performance is greatly improved over the traditional one-stage and two-stage frameworks, especially based on real-time conditions; at the same rate, its accuracy should be based on deep learning motion. Human target recognition and pose estimation are much higher than YOLOV3 [18], and compared to YOLOV3, it can recognize small objects significantly better. From the above introduction, we know that the two-stage detector first detects all potential object positions (i.e., candidate areas) for each category to judge, and this process consumes a lot of time and storage space, which means it is not applicable for real-time object detection.

Foreground target detection mainly serves advanced video analysis technology in intelligent video surveillance. The main task is to extract interesting foreground targets in real surveillance scenes. Beaugendre et al. [19] proposed a random block background modeling (RBBM) algorithm. The update of the background model is divided into multiple random blocks evenly distributed on the image and in time so as to quickly update the video background image. This makes it possible to save a lot of calculation time when processing high-definition video images while sacrificing certain detection accuracy requirements. Subsequently, Beaugendre et al. [20] proposed a mixed block background modeling (MBBM) algorithm based on the spatiotemporal update. The background model is updated by carefully selecting blocks in linear and pseudorandom order and updating the block part of the corresponding model. The two-block selection sequence ensures that each block will be updated. This algorithm is combined with the adaptive block propagation background subtraction method (ABPBGS) [21] for foreground detection. Savas et al. [22] proposed an algorithm based on block matching to estimate the motion in each frame and generated the corresponding motion field to serve the foreground target detection, which showed a better detection effect in the crowded situation. In order to overcome the shortcomings of convolutional neural networks in foreground detection, Dimitrios et al. [23] proposed an end-to-end 3D convolutional neural network model and used it for foreground target detection. The model can track changes in time, so there is no need to retain and update the model.

3. Methodology

3.1. Confidence Weighted Fusion and Visual Attention. For the foreground processing of football players' competitive video, this paper proposes a foreground detection method based on confidence weighted fusion and visual attention to solve the problem of color camouflage and static foreground in the background subtraction method. The algorithm first builds a model through color features, LBSP texture features, and the corresponding confidence, and then calculates the sum of the confidence of valid samples for weighted fusion for pixel classification and builds a visual attention mechanism to determine the static foreground. Finally, the model samples are updated with the minimum sample confidence strategy, and a two-dimensional confidence update strategy and an adaptive weight update strategy are constructed to update the corresponding confidence and weight.

3.1.1. Pixel Classification for Sample Consistency. The pixel classification method based on sample consistency is based on statistics to determine whether the current pixel is the foreground by comparing the current frame with the samples in the model. In the actual activity of pixel classification, two different pixel matching methods are shown according to the difference of the background model.

(1) Color dimension pixel classification: In a background model composed of individual color features such as the ViBe algorithm and the PBAS algorithm, the samples in the model are composed of n color values, as shown in the following equation:

$$B(x) = \{v_1(x), v_2(x), ..., v_n(x)\}. \tag{1}$$

Therefore, only the pixel classification of the color dimension is required in the foreground detection stage. The pixel classification method is shown in Figure 1. The pixel point $I_t(x)$ of the current frame is taken as the center, and the color value of the sample in the model in a circle with a radius of $R(x)$ is compared, and the number of samples in the circle is recorded. If it is greater than or equal to the minimum matching number min, the current pixel can be judged to be the background; otherwise, it is the foreground. The equation is as follows:

$$F_t(x) = \begin{cases} 1, & \text{if} \#\{\text{dist}(I_t(x), B_n(x)) < R(x), \forall n\} < \min, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $F_t(x) = 1$ indicates that the current pixel is the foreground; otherwise it is recorded as the background. $\#\{\ldots\}$ represents the number of samples whose Euclidean distance between the observation and the sample is less than the given distance threshold. min is the minimum number of matches that satisfy the condition.

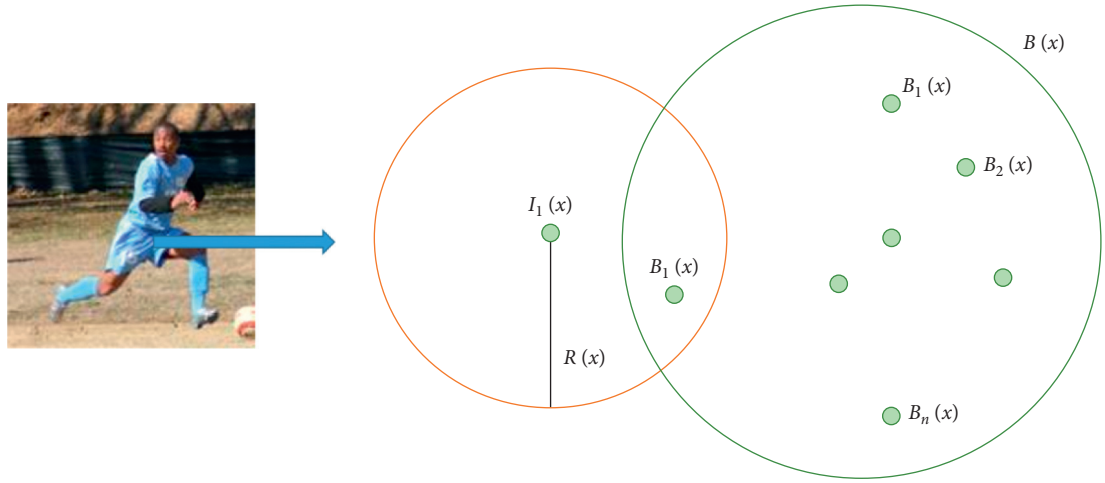

Figure 1: Example of color dimension pixel classification.

(2) Joint verification: Different from the background model composed of a single color value, the pixel-level model constructed by combining the color value and the texture value (taking the LBSP texture feature value as an example) uses the joint verification method of color dimension and texture dimension when performing foreground detection. For pixel classification, the representative ones are LOBSTER algorithm, SuB SENSE algorithm, etc. As shown in Figure 2, the pixel classification of the color dimension is first performed during pixel classification. If it is preliminarily determined as the foreground in the color dimension, the pixel classification of the texture dimension is performed; otherwise, the pixel classification activity of the texture dimension is not performed. If the logical AND operation value is 1, it means that a match is obtained; otherwise, it is not considered a match.

$$\left(\operatorname{dist}\left(v_t(x), v_i(x)\right) < R_c(x)\right) \&\& \left(\operatorname{dist}\left(\operatorname{lbsp}_t(x), \operatorname{lbsp}_i(x)\right) < R_{L1}(x)\right). \tag{3}$$

As can be seen from Figure 2, the pixel classification of the texture dimension is consistent with the pixel classification of the color dimension. The difference is that the pixel classification of the texture dimension uses the Hamming distance as a measure of similarity distance. Predict and determine whether the current pixel is the foreground or background by judging whether the number of matches obtained is less than the minimum matching value min. Taking the color feature and LBSP texture feature to construct the background model as an example, the pseudocode implementation process based on the double verification method of color level and texture level is shown in Algorithm 1.

3.2. Pixel Classification Algorithm Based on Confidence Weighted Fusion. In the model initialization stage, the background model $B(x)$ is established by obtaining the pixel information of the previous frame. The model is composed of N modules, and the equation is as follows:

$$B(x) = \{B_1(x), B_2(x), \ldots, B_i(x), \ldots, B_n(x)\}. \tag{4}$$

Different from the pixel-level model of regular sample consistency, the template $B_i(x)$ consists of color value v_i, LBSP texture feature value $\mathrm{LBSP}_i(x)$, color level confidence level $C_i^1(x)$, and texture level confidence center $C_i^2(x)$ composition. The calculation equation is as follows:

$$B_i(x) = \left\{v_i, \mathrm{LBSP}_i(x), C_i^1(x), C_i^2(x)\right\}. \tag{5}$$

In the foreground segmentation, that is, when pixel classification is performed, the samples whose distance between the current pixel $I_t(x)$ and the sample in the model is less than the given distance value $R(x)$ are recorded as strong correlation samples, and the number is obtained n. Mark the color confidence $C_i^1(x)$ and texture confidence $C_i^2(x)$ corresponding to the strong correlation samples, denoted as $t_i^1(x)$ and $t_i^2(x)$, respectively. The calculation equation is as follows:

$$t_i^m(x) = \begin{cases} C_i^m(x), & \operatorname{dist}\left(I_t(x), B_i(x)\right) < R(x), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where m takes the value 1 or 2, corresponding to the color dimension and texture dimension. Euclidean distance is used for color dimension judgment, and Hamming distance is used for texture dimension judgment. Then, the color confidence and texture confidence of the strong correlation samples are, respectively, summed, and then weighted and

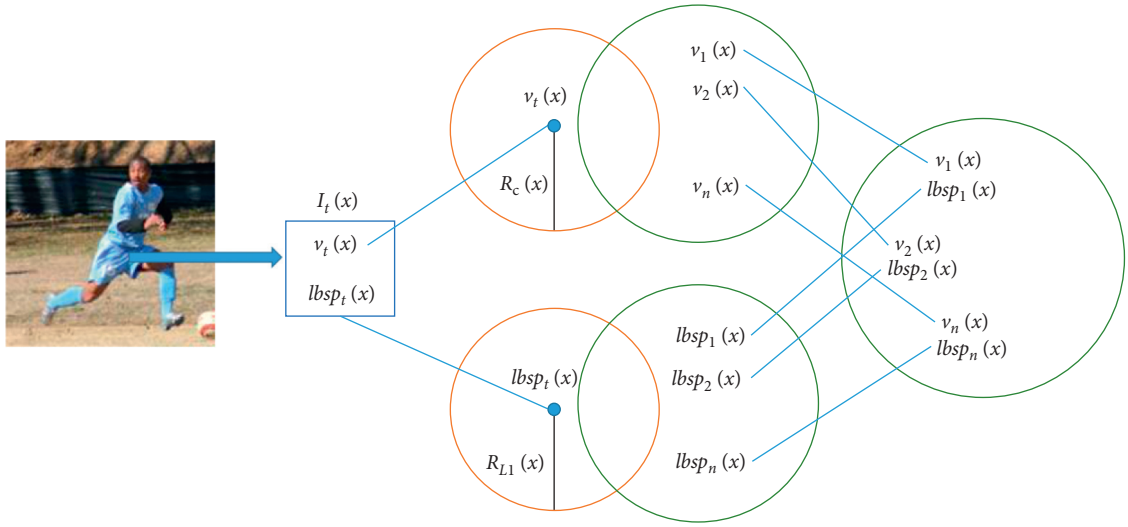

Figure 2: Example of joint verification pixel classification.

Input: current input frame pixel $I_t(x)$
Output: the foreground/background label of $I_t(x)$
(1) initialize related variable (nCounts = 0, i = 0)
(2) while nCounts < min && i < N
(3) colorDist = dist($I_t(x)$, $v_i(x)$)
(4) if colorDist ≥ R_{color}
(5) goto wrongMatch;
(6) lbspDist = dist(lbsp$_t(x)$, LBSP$_i(x)$)
(7) if lbspDist ≥ R_{color}
(8) goto wrongMatch;
(9) nCounts++;
(10) wrongMatch;
(11) i++;
(12) if nCounts < min
(13) $I_t(x)$ is foreground
(14) else
(15) $I_t(x)$ is background

Algorithm 1: Pixel classification for sample consistency.

summed. If it is less than the minimum threshold min, it is judged as the foreground; otherwise, it is the background.

$$F(x) = \begin{cases} 1, & \lambda_1(x)\sum_{i=1}^{n} t_i^1(x) + \lambda_2(x)\sum_{i=1}^{p} t_i^2(x) \le \min, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

3.3. DetectNet. This section describes the specific implementation of DetectNet. Due to the outdated technology and network framework of the network architecture provided by the original text [17], it cannot meet the requirements of the overall framework of the system in this article. It is necessary to reimplement CenterNet, including training and testing of the network architecture. The input image is $I \in R^{W*H*3}$, and the target output is the heat map of generating key points $\widehat{Y} \in [0,1]^{(W/R)*(H/R)*C}$ where R is the transformation scale and C is the number of keypoint output feature channels (that is, the number of categories), and three basic frameworks are needed to generate the heat map: ResNet (including ResNet101, ResNet18, etc.), stacked hourglass network, and deep layer aggregation (DLA). As shown in Figure 3, (a) hourglass network, we use it just like in CornerNet; (b) ResNet and transposed convolution. Add a 3×3 deformable convolutional layer before each upsampling layer. Specifically, first, use deformable convolution to change the channel and then use convolution to upsample the feature map (the two steps are, respectively, in $32 \longrightarrow 16$. This article shows these two steps together as $16 \longrightarrow 8$ and $8 \longrightarrow 4$ dotted arrows); (c) use the original DLA-34 for semantic segmentation; (d) about DLA-34. Add more skip

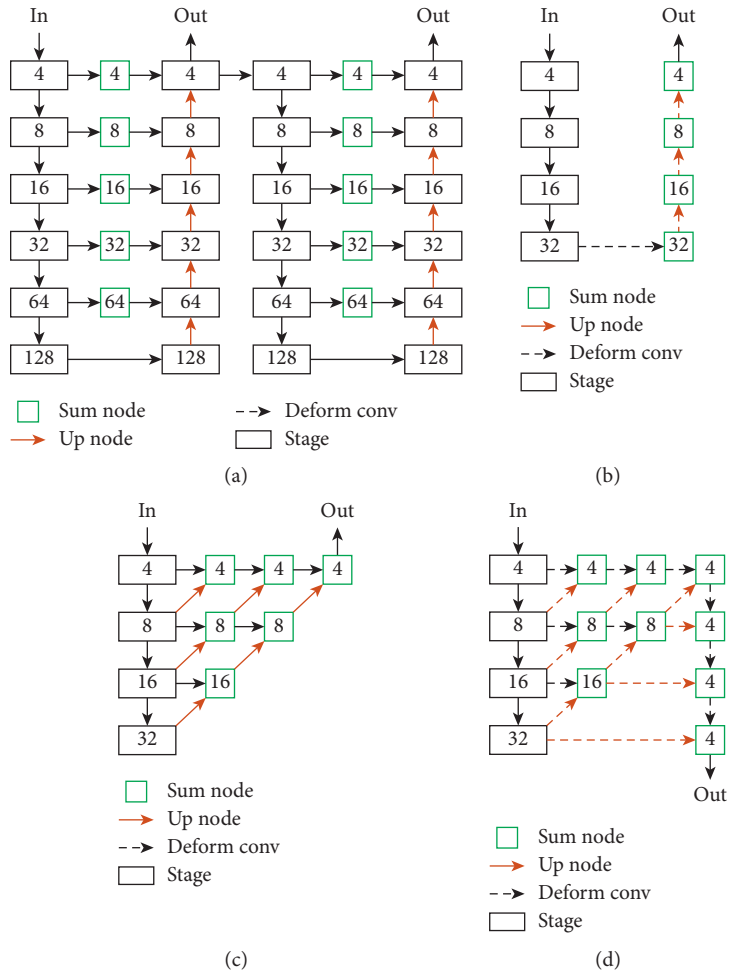

Figure 3: Proposed DetectNet.

connections at the bottom layer and upgrade each convolutional layer into the upsampling stage to a deformable convolutional layer. Combining the performance of each network given in the paper and our test results, taking into account the needs of this paper, realize the ResNet-based DetectNet network.

The objective function of the training is a pixel-level logistic regression. For the key point p of each category c, first, calculate a low-resolution equivalent value $\widetilde{P} = [P/R]$ and then calculate the key on the heat map through the Gaussian kernel. The point is as follows:

$$Y_{xyc} = \exp\left(-\frac{\left(x - \breve{p}_x\right)^2 + \left(y - \breve{p}_y\right)^2}{2\sigma_p^2}\right), \tag{8}$$

$$L_k = \frac{1}{N}\sum_p \begin{cases} \left(1 - \widehat{Y}_{\text{xyc}}\right)^{\alpha} * \log\left(\widehat{Y}_{\text{xyc}}\right) \text{if } Y_{\text{xyc}} = 1 \\ \left(1 - \widehat{Y}_{\text{xyc}}\right)^{\beta} * \left(\widehat{Y}_{\text{xyc}}\right)^{\alpha} \\ \log\left(\widehat{Y}_{\text{xyc}}\right) \end{cases}, \tag{9}$$

where α and β are the hyperparameters of focal loss, respectively, and N is the number of key points in the image.

When $Y_{xyc} = 1$, if $\widehat{Y}_{xyc}$ is close to 1, it means that this is very easy to detect point, and the corresponding $(1 - \widehat{Y}_{\text{xyc}})^{\alpha}$ is very low.

When $Y_{xyc} = 1$, if $\widehat{Y}_{xyc}$ is close to 0, then it means that this is a point that is not easy to detect. That is, the center point has not been learned, so the proportion of training should be increased, and the corresponding $(1 - \widehat{Y}_{\text{xyc}})^{\alpha}$ is very big.

4. Experiments and Results

4.1. Experimental Environment. Since the experiment in this article needs to train a deep neural network [24, 25], the scale is large, the structure is more complex, and the calculation scale is large. The programming language used is Python, the version is 3.6, the deep learning framework used is Keras2.1.5, and the IDE for program deployment is Pycharm, and all experiments are conducted in the same environment. All our experiments have been conducted on a desktop PC with an Intel Core i7-8700 processor and an NVIDIA GeForce GTX 1080ti GPU.

4.2. Datasets Preprocessing

4.2.1. 2D Pose Estimation Datasets. The training and verification of the 2D Pose network need to use some public data sets. Compared with the 3D human pose estimation, since the pictures we need can be obtained directly from the Internet, such as YouTube, Flickr, etc., the 2D human pose data set is relatively rich. In addition, these data sets have high-quality pictures while providing the two-dimensional coordinates of the skeleton of the characters in their images. These data sets not only provide image data containing the human body but also provide the annotation information corresponding to the pose of the human body, which is convenient for training your own 2D pose estimation network and has a unified standard evaluation and comparison of the performance of each network, generally using mAP as the evaluation standards; the following are some commonly used public data sets.

(1) MPII dataset, used as a benchmark for evaluating hinged human skeleton: The data set has about 25,000 high-quality images in total, of which more than 40,000 high-quality images containing people are annotated with two-dimensional skeleton coordinates, and the number of skeleton key points is 16. These images are derived from human daily life, collected by the classifier. There are about 410 different human activities, and they contain their corresponding two-dimensional skeleton coordinate labels; at the same time, most of these images are extracted from online videos, such as YouTube, so it also provides a lot of frame images without label information, which is very useful in the testing and training of some methods; at the same time, its test set contains the label information of the three-dimensional torso and the direction after the body is occluded.

(2) MSCOCO dataset, the full name is Miscrpspft COCO Dataset: we have already introduced the detection data set, so this data set is very large and has a wide range of uses. It can be used for detection and tracking, segmentation, and two-dimensional pose estimation. Here we only use the key point information of the human body's two-dimensional skeleton for two-dimensional pose estimation. As far as the MSCOCO data set contains the joint point information of the human body skeleton, this part includes more than 200,000 pieces of various types derived from the network. Daily pictures: There are more than 250,000 different people doing various actions in all pictures, and the skeleton coordinates are marked for these different people. The number of skeleton key points is 18, so we can see the data set. The amount of data far exceeds that of MPII.

(3) LSP dataset: The full name of the LSP data set is Leeds Sports Pose Dataset. This data set has a small amount of data. The overall data set contains about two thousand images of skeleton annotation information. Most of these images in the data set are from the data set obtained by Filcker which is mainly for the data set related to the task of moving target characters. Therefore, all images are required to contain not only the annotation information of the human skeleton but also the category labels of various sports, which is convenient to distinguish different sports categories and the number of key points of the human skeleton Is 14.

In the process of training the model, this article performs preprocessing such as enhancement to the data set, removing the traditional rotation of the picture by a certain angle, zooming in a certain proportion, etc.; this article uses a data enhancement processing method different from the above method; here, we call this data standardization. Reference [53] proposed a new standardization method for data, such as formula (10), to find a λ to minimize it.

$$\operatorname{argmin}_{\lambda} \left\| \lambda P_c - P_u \right\|^2. \tag{10}$$

P_c is the x, y coordinate value of the two-dimensional skeleton coordinate point, and P only takes the x, y coordinate value of the corresponding three-dimensional skeleton coordinate. λ is obtained by this method, and the accuracy obtained in the final training is significantly improved. This article studies several methods to solve the vector λ size is $N \times 1 \times 2$, where N generally corresponds to batch size (the value in this experiment is 1024).

4.2.2. 3D Human Pose Estimation Datasets. Because the production of 3D human skeleton data sets is very difficult and complicated, its collection requires a large number of cameras, sensors, and the full support of depth cameras to ensure the accuracy of the marked 3D coordinates. Therefore, many 3D skeleton data sets are indoors. It is difficult to have outdoor data, which leads to the lack of natural data to make the network learn better. However, the corresponding two-dimensional skeleton data is indeed diverse and accurate, containing a large amount of indoor and outdoor various types of data. This is also the reason why the existing two-dimensional pose estimation methods are more stable and accurate. As a result, many advanced three-dimensional pose estimations use two-dimensional data. Based on pose estimation, 3DPoseNet is also based on inputting two-dimensional skeleton key points to predict and output corresponding three-dimensional skeleton key points.

Human3.6M dataset is the main evaluation data set of existing advanced paper methods. This article is mainly based on this data set evaluation. There are 3.6 million images in the data set. There are 11 experimenters (6 males and 5 females) in the data, that is, 11 different characters. Each image contains the corresponding 2D skeleton and 3D skeleton coordinates. The data is captured by 4 digital cameras, 1 time sensor, and 10 sports cameras. It contains 17 action scenes such as discussion, eating, calling, sports, sitting, standing, walking, and greeting, etc. The number of key points of the skeleton is 17. Usually, data S1, S5, S6, S7,

(a) (b)

Figure 4: The 2D pose detection result on the LSP dataset.

S8 are used as the training set, while S9, S11 are used as the corresponding test set.

4.3. DPoseNet Experimental Results. According to the above experiments on many two-dimensional pose estimation methods, an independent and clear target person processed by 2DPoseNet (2DPN) will be obtained. Essentially, it is the process of transforming the two-dimensional human pose coordinates to the three-dimensional human pose coordinates, that is, improving the dimensionality. In this process, it is necessary to make full use of the position information between the key points of each joint of the human body. And 2DPoseNet generates 2D key points funny and accurately, especially for self-occlusion. The method belongs to the top-down method, so it has certain limitations. For example, when there are too many people in the multiperson 2D human pose estimation, our method will have certain defects in speed and accuracy. The position of each joint point of the human body is marked from the image, and each estimated position contains only a two-dimensional coordinate, so the final output of the two-dimensional human body pose estimation method is a two-dimensional human body pose. The position coordinates of each joint of the human body are drawn according to the image. That is, each estimated position corresponds to only one two-dimensional coordinate. Regarding our 2DPoseNet model training, we still use traditional data set training, such as LSP.

As shown in Figure 4 and Tables 1–3 , the single-person 2D human pose estimation is very accurate in the complex environment.

4.4. DPoseNet Experimental Results. This section will show the training process of the 3DPoseNet network. That is, to train the 3DPoseNet in this article, the initial learning rate used is $lr = 0.001$, and each cycle experiences exponential decay. Define dropout $p = 0.25$. For the convolutional layer, set the convolution kernel $W = 1$, and set it to $C = 1024$ output channels. Compared with the previous algorithm, we can see that the accuracy of the final 2D pose skeleton obtained by the method in this paper has been significantly improved. Based on the Human3.6M data set, as shown in Table 4, it is the result of comparing the advanced methods in this paper. Regarding the evaluation methods from 2D human pose estimation to 3D human pose estimation, this paper uses a series of 2D keypoint sequences as input and uses the same evaluation indicators as related papers on 3D pose estimation, namely P_MPJPE. Procrustes analysis MPJPE (P_MPJPE): Denoted as Protocol 2, which is MPJPE based on Procrustes analysis. After rigid transformations such as scaling, translation, and rotation are performed on the network output, the MPJPE is calculated after aligning to the true value.

As shown in Table 4, it can be clearly found that the accuracy of this method is greatly improved.

As shown in Figures 5 and 6 , the visual results of the single-person pose predicted by the method in this paper are all single-person 3D human pose renderings based on 30fps video processing in this paper. As shown in Figure 6, it is the effect of the multiperson 3D pose predicted by the method in this paper. The method in this paper can basically accurately predict the movements of the multiperson movement.

Table 1: Average precision of joint detection on LSP.

LSP	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Neck	Head	mAP
Wang et al. [26]	0.25	0.34	0.23	0.23	0.43	0.32	0.21	0.34	0.31
Pishchulin et al. [27]	0.37	0.31	0.36	0.31	0.43	0.33	0.32	0.76	0.36
Ours	**0.47**	**0.41**	**0.49**	**0.39**	**0.44**	**0.52**	**0.59**	**0.80**	**0.49**

Table 2: Average precision of joint detection on MSCOCO.

MSCOCO	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Neck	Head	mAP
Wang et al. [26]	0.20	0.31	0.21	0.19	0.39	0.31	0.19	0.31	0.30
Pishchulin et al. [27]	0.32	0.28	0.32	0.33	0.42	0.36	0.31	0.72	0.33
Ours	**0.42**	**0.39**	**0.45**	**0.34**	**0.43**	**0.49**	**0.47**	**0.77**	**0.41**

Table 3: Average precision of joint detection on MPII.

MPII	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Neck	Head	mAP
Wang et al. [26]	0.31	0.30	0.25	0.28	0.39	0.33	0.27	0.32	0.34
Pishchulin et al. [27]	0.39	0.33	0.32	0.30	0.43	0.39	0.33	0.74	0.33
Ours	**0.42**	**0.46**	**0.43**	**0.37**	**0.48**	**0.51**	**0.49**	**0.78**	**0.45**

Table 4: The results of the Mean Per Joint Position Error(MPJPE), with less values, mean better results.

MPII	Direct.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit
Martinez et al. [28]	0.39	0.43	0.46	047	0.51	0.56	0.41	0.40	0.56
Sun et al. [29]	0.42	0.44	0.45	0.45	0.51	0.53	0.43	0.41	0.59
Fang et al. [30]	0.38	0.41	0.43	0.44	0.48	0.55	0.40	0.38	0.54
Pavlakos et al. [31]	0.34	0.39	0.41	0.38	0.42	0.47	0.38	0.36	0.50
Yang et al. [32]	0.26	0.30	0.39	0.39	0.43	0.47	0.28	0.29	0.36
Hossain et al. [33]	0.35	0.39	0.43	0.43	0.47	0.54	0.38	0.37	0.51
Pavllo et al. [34]	0.34	0.36	0.37	0.37	0.36	0.42	0.34	0.33	0.45
Ours	**0.26**	**0.31**	**0.28**	**0.29**	**0.32**	**0.38**	**0.29**	**0.27**	**0.36**

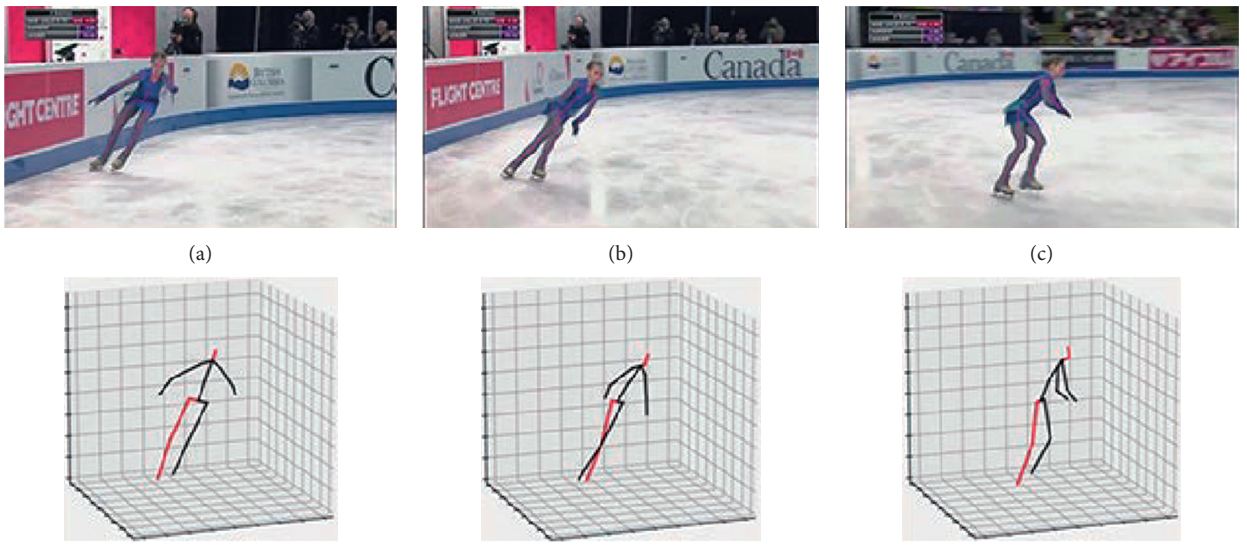

Figure 5: The single-person 3D pose detection results.

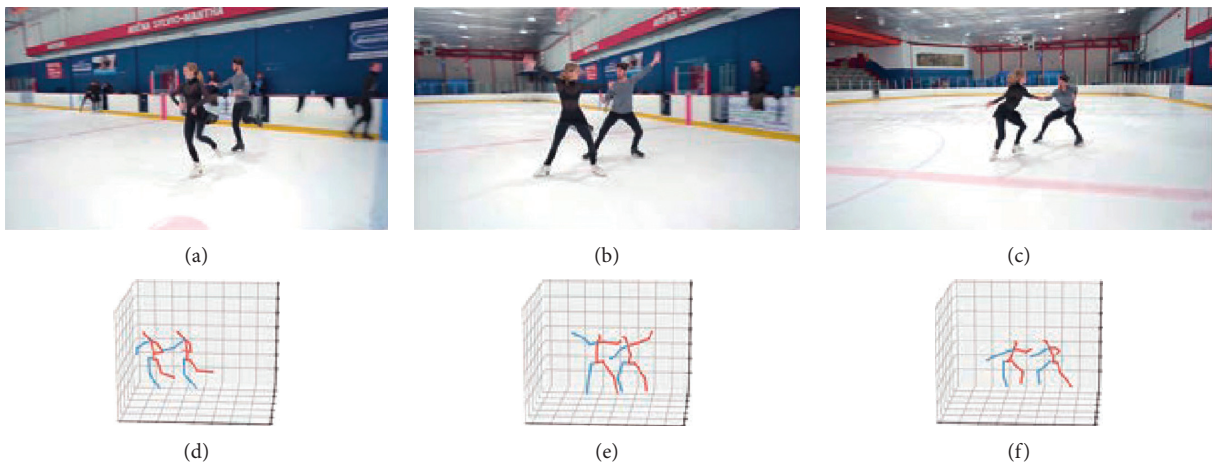

Figure 6: The Multiperson 3D pose detection results.

5. Conclusion

Since football player's posture detection has important application value in sports competition for different purposes of analysis. Therefore, this paper proposes a football motion detection approach combining foreground detection and deep learning for real-time detection of football players' posture. The main task of foreground target detection is to extract the foreground target of interest in the real surveillance scene and use it as the target of interest for subsequent analysis. Then, we propose a triple DetectNet detection framework based on deep learning technology, which can quickly and reliably realize the three-dimensional pose estimation of multiperson motion. For input, the triple DetectNet framework uses three neural networks and is executed in three stages; the first stage is to use the DetectNet network to detect the bounding box of each person separately, the second stage uses the 2DPoseNet network to estimate each person's response, and the third stage uses the 3DPoseNet network to obtain the person's 3D pose. The validity of the proposed approach has been tested by conducting several experiments on four data sets, and the results proved the success and superiority of the algorithm.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author does not have any possible conflicts of interest.

References

[1] G. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[2] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, and S. Tian, "Feature refinement and filter network for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[3] W. Cai and Z. Wei, "PiiGAN: generative adversarial networks for pluralistic image inpainting," *IEEE Access*, vol. 8, pp. 48451–48463, 2020.

[4] Y. Tong, L. Yu, S. Li, J. Liu, H. Qin, and W. Li, "Polynomial fitting algorithm based on neural network," *ASP Transactions on Pattern Recognition and Intelligent Systems*, vol. 1, no. 1, pp. 32–39, 2021.

[5] W. Cai and Z. Wei, "Remote sensing image classification based on a cross-attention mechanism and graph convolution," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[6] X. Ning, Y. Wang, W. Tian, L. Liu, and W. Cai, "A biomimetic covering learning method based on principle of homology continuity," *ASP Transactions on Pattern Recognition and Intelligent Systems*, vol. 1, no. 1, pp. 9–16, 2021.

[7] W. Cai, Z. Wei, R. Liu, Y. Zhuang, Y. Wang, and X. Ning, "Remote sensing image recognition based on multi-attention residual fusion networks," *ASP Transactions on Pattern Recognition and Intelligent Systems*, vol. 1, no. 1, pp. 1–8, 2021.

[8] X. Zhang, Y. Yang, Z. Li, X. Ning, Y. Qin, and W. Cai, "An improved encoder-decoder network based on strip pool method applied to segmentation of farmland vacancy field," *Entropy*, vol. 23, no. 4, p. 435, 2021.

[9] Z. Chu, M. Hu, and X. Chen, "Robotic grasp detection using a novel two-stage approach," *ASP Transactions on Internet of Things*, vol. 1, no. 1, pp. 19–29, 2021.

[10] R. Liu, X. Ning, W. Cai, and G. Li, "Multiscale dense cross-attention mechanism with covariance pooling for hyperspectral image scene classification," *Mobile Information Systems*, vol. 2021, 2021.

[11] Z. L. Yang, S. Y. Zhang, Y. T. Hu, Z. W. Hu, and Y. F. Huang, "VAE-Stega: linguistic steganography based on variational auto-encoder," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 880–895, 2020.

[12] X. Ning, X. Wang, S Xu et al., "A review of research on co-training," *Concurrency and Computation: Practice and Experience*, 2021.

[13] A. Keivani, J. R. Tapamo, and F. Ghayoor, "Motion-based moving object detection and tracking using automatic K-means," in *Proceedings of the 2017 IEEE AFRICON*, pp. 32–37, IEEE, Cape Town, South Africa, September 2017.

[14] K. E. Van De Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Proceedings of the 2011 International*

Conference on Computer Vision, pp. 1879–1886, IEEE, Barcelona, Spain, November 2011.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[16] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[17] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, https://arxiv.org/abs/1904.07850.

[18] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[19] A. Beaugendre and S. Goto, "Block-propagative background subtraction system for UHDTV videos," *Information and Media Technologies*, vol. 10, no. 2, pp. 259–262, 2015.

[20] A. Beaugendre, S. Goto, and T. Yoshimura, "Real-time UHD background modelling with mixed selection block updates," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, no. 2, pp. 581–591, 2017.

[21] A. Beaugendre and S. Goto, "Adaptive block-propagative background subtraction method for UHDTV foreground detection," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98.A, no. 11, pp. 2307–2314, 2015.

[22] M. F. Savaş, H. Demirel, and B. Erkal, "Moving object detection using an adaptive background subtraction method based on block-based structure in dynamic scene," *Optik*, vol. 168, pp. 605–618, 2018.

[23] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3D convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 23023–23041, 2018.

[24] C. Yan, G. Pang, X. Bai, J. Zhou, and L. Gu, "Beyond triplet loss: person Re-identification with fine-grained difference-aware pairwise loss," *IEEE Transactions on Multimedia*, 2021.

[25] C. Wang, X. Bai, X. Wang et al., "Self-supervised multiscale adversarial regression network for stereo disparity estimation," *IEEE Transactions on Cybernetics*, 2020.

[26] F. Wang and Y. Li, "Beyond physical connections: tree models in human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 596–603, Portland, OR, USA, June 2013.

[27] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588–595, Portland, OR, USA, June 2013.

[28] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2640–2649, Venice, Italy, October 2017.

[29] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2602–2611, Venice, Italy, October 2017.

[30] H. S. Fang, Y. Xu, W. Wang, X. Liu, and S. C. Zhu, "Learning pose grammar to encode human body configuration for 3D pose estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[31] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3D human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7307–7316, Salt Lake City, UT, USA, June 2018.

[32] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3D human pose estimation in the wild by adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5255–5264, Salt Lake City, UT, USA, June 2018.

[33] M. R. I. Hossain and J. J. Little, "Exploiting temporal information for 3d human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 68–84, Munich, Germany, September 2018.

[34] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7753–7762, Long Beach, CA, USA, June 2019.