Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3326847, 14 pages
https://doi.org/10.1155/2021/3326847

Research Article

Sports Sequence Images Based on Convolutional Neural Network

Yonghao Chen

Sports Department, Xi'an Medical University, Xi'an 710021, Shaanxi, China

Correspondence should be addressed to Yonghao Chen; chenyonghao@xiyi.edu.cn

Received 27 April 2021; Revised 28 May 2021; Accepted 4 June 2021; Published 24 June 2021

Academic Editor: Ming Bao Cheng

Convolution neural network has become a hot research topic in the field of computer vision because of its superior performance in image classification. Based on the above background, the purpose of this paper is to analyze sports sequence images based on convolutional neural network. In view of the low detection rate of single-frame and the complexity of multiframe detection algorithms, this paper proposes a new algorithm combining single-frame detection and multiframe detection, so as to improve the detection rate of small targets and reduce the detection time. Based on the traditional residual network, an improved, multiscale, residual network is proposed in this paper. The network structure enables the convolution layer to "observe" data from different scales and obtain more abundant input features. Moreover, the depth of the network is reduced, the gradient vanishing problem is effectively suppressed, and the training difficulty is reduced. Finally, the ensemble learning method of relative majority voting is used to reduce the classification error rate of the network to 3.99% on CIFAR-10, and the error rate is reduced by 3% compared with the original residual neural network.

1. Introduction

Convolutional neural network is a special multilayer perceptron designed to recognize two-dimensional images that can automatically extract image features. The original image does not require a lot of preprocessing to better learn the invariance characteristics of the image. Now, the typical convolutional neural network is a multilayer trainable architecture, including input, convolutional layer (local connection layer), sampling layer, normalization layer, fully connected layer, logistic regression layer, output layer, and the like. Methods to improve the image recognition effect of convolutional neural networks, methods to find the network structure and parameter configuration that are most suitable for the data set to be recognized, and the network structure with constant compatibility for various data sets have gradually become the hotspots of current research.

Due to the importance of motion sequence image research, many research teams have begun to study motion sequence images and have achieved good results. Yang and Jiang studied the tracking theory based on the Candide3 face model and further developed the tracking process on this basis. Based on the research of face tracking, a dynamic feature extraction method based on six parameters of face model is proposed. The facial expression feature point location and tracking algorithm based on active appearance model was introduced, and the tracking principle was studied. On the basis of the Candide3 face model, the tracking process is further developed. Dynamic time warping (DTW) technology was used to align the image sequence, and then, the feature vector was extracted [1]. In recent years, a database containing a large number of human motion patterns has appeared and is expected to be reused as a new method for recognizing motions and recovering motion patterns from videos containing monocular images. Guermazi and Roemer described the design of an action database, which is composed of action configuration, pose descriptors in silhouette images, random models coded by each pose descriptor subsequence, and the relationship between data and random models. The proposed action database is used to identify images that contain specific actions performed by performers and recover all joint angles from the images [2]. Although the current research results are relatively rich, there are still shortcomings, mainly reflected in the huge amount of data and cumbersome analysis time.

In the research of algorithm analysis, convolutional neural network is a very good method that can solve many classification problems, so it is widely used in the research of algorithm analysis. He proposed the first convolutional neural network (CNN) that provide real-time SR 1080p video on a k2gpu. To achieve this, they proposed a new CNN architecture to extract feature maps in the LR space. In addition, they also introduced an efficient subpixel convolutional layer, which learns a set of upwardly expanded filters to upgrade the final LR feature map to the HR output. By doing so, they effectively replaced the bicubic filter in the hand-made SR pipeline, trained more complex upscaling filters for each feature map, and reduced the computational complexity of the entire SR operation [3]. Dong et al. proposed an online visual tracking algorithm that uses convolutional neural network (CNN) to learn and discriminate saliency maps. Assuming that CNN is pretrained on a large-scale offline image library, their algorithm uses the output of the hidden layer of the network as a feature descriptor because they show good performance in various common visual recognition problems. These features are used for online support vector machine (SVM) to learn and recognize the target appearance model [4, 5]. Due to the effectiveness of the image analysis method, the convolutional neural network method can be applied to the analysis of sports sequence images to solve the problem of slow image analysis.

In this paper, motion sequence image target detection and tracking system based on a convolutional neural network is established. A simple convolutional neural network model is established according to the image characteristics. When selecting hardware, the software system can be used to select appropriate parameters and best parameters in combination with specific conditions. The optimal algorithm can not only reduce the burden when selecting the algorithm but also provide powerful parameters for the experimental program, without wasting too much time, and can be used to complete the image classification training process and display the classification results.

2. Convolutional Neural Network and Sequence Image

2.1. Convolution Neural Network Learning Algorithm

2.1.1. The Training Process of Network. The training process of convolutional neural network is similar to the traditional BP algorithm, including four steps, which is divided into two stages:

The first stage: forward propagation process:

(1) Take a sample from the sample set and input it into the network

(2) Calculate the corresponding actual output

In this stage, the input information will be transformed according to the hierarchy and output to the output layer. In the calculation process through the network, in order to obtain the final output result, a little deviation is added in the weight column of input and each output.

The second stage: the process of back propagation:

(1) Calculate the difference between the actual output and the expected output

(2) The weight matrix is propagated and adjusted according to the error minimization method

The network training process includes net propagation and antipropagation [6, 7]. The forward wave mainly includes the feature extraction and classification calculation. The inverse wave is the wrong inverse feedback and updates calculation of weight value. After inputting images, all output must be initialized. Exhibition and sampling realize the extraction and mapping of image features, multiple exhibition and sampling processes can be used here [8]. The multilayer extraction process can extract useful information from the image. After feature extraction, the extracted features will be conveyed to the fully connected hierarchy again. There are several hidden layers in the fully connected layer. The results are fed back to the output layer through the conversion and calculation of the data information in the hidden layer. The output layer makes some calculations and gets the test results. Compare the test results with the expected results, and output the classified results if they are consistent [9, 10].

In backpropagation, for a training sample (x, y), the loss function is defined as shown in

$$S(w, b, x, y) = \frac{1}{2}\left\| g_{w,b}(x) - y \right\|^2, \tag{1}$$

where y is the real result and g is the predicted output of the neural network. For training data containing n samples, the overall loss function is shown in

$$S(w, b) = \frac{1}{n}\sum_{i=1}^{n} S\left(w, b, x^{(i)}, y^{(i)}\right). \tag{2}$$

Generally speaking, the loss function of a neural network is a nonconvex function and often converges to a local minimum. The gradient descent method can be used to update the parameters. The main goal is to find the partial derivative of the objective function with respect to the parameter vector. The solution formula is shown in

$$w_{i,j}^{(I)} = w_{i,j}^{(I)} - \varepsilon \frac{\partial}{\partial w_{i,j}^{(I)}} J(w, b), \tag{3}$$

$$b_i^{(I)} = b_i^{(I)} - \varepsilon \frac{\partial}{\partial b_i^{(I)}} J(w, b), \tag{4}$$

where ε is the learning rate.

If the test result is not consistent with the expected result, the weight value and deviation should be propagated again from the output layer to the fully connected layer and layering, until each output has its own gradients and then carries out weight updating work to start a new training process [11, 12]:

(1) *Convolution and Sampling Process*. The convolution process is the process of using a template to perform a template operation on an image. The template can also be a filter or a convolution kernel [13, 14]. The template calculation formula is shown in

$$F(x, y) * W(g, h) = \sum_{i=c}^{c} \sum_{j=d}^{d} w(i, j) f(x + i, y + j). \quad (5)$$

Among them, $*$ is the convolution operator, $F(x, y)$, $f(x, y)$, respectively, represent the domain image and the pixel value centered on the pixel point (x, y). $W(g, h), w(g, h)$ represent the template matrix and the weight in the matrix, respectively.

The specific process of the first integration process: first input image, then feature extraction process. The process of feature extraction is to integrate the filters, which can learn the input image, and then add bias. The protocol layer is generated after feature extraction. Then, the adjacent regions of the conformity layer are maximized or averaged [15]. In this process, the corresponding weight value and deviation need to be added, and the output can be obtained by activating the function. The results generate a sampled function map. In the later convolution process, the input of convolution level becomes the output of sampling layer in the previous convolution process [16, 17]. Commonly used activation functions include Sigmoid function, Tanh function, and Relu function. The formulas are as shown in

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (6)$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \quad (7)$$

$$f(x) = \max(0, x). \quad (8)$$

In order to prevent the difference between adjacent features in the same channel of the feature image from being too large, the network will normalize the features when extracting the features. The normalized formula is

$$g_{x,y}^{i} = \frac{h_{x,y}^{i}}{\left(r + \alpha \sum_{j=\max(0, i - n/2)}^{\min(N-1, i+n/2)} \left(h_{x,y}^{j}\right)^{2}\right)^{\beta}}. \quad (9)$$

Among them, $g_{x,y}^{i}, h_{x,y}^{i}$ are the original activation and new activation of the convolution kernel, respectively; N, n are the number of current network layer convolution kernels and the number of channels of the feature image, respectively.

(2) *Normalization Process*. The main function of normalization layer is to extract features after sampling layer. Local normalization is divided into local normalization and local contrast normalization in different functional maps [18]. In the process of feature extraction in the first stage, the local normalization on the same feature graph is added between convolution and subsampling. In the second stage, local normalization between different features and graphs is added. This layer is very useful when we use unbounded neuronal activation functions [19, 20]. Because it makes neurons produce larger response, local areas allow the detection of high-frequency characteristics, resulting in fierce competition of local neurons.

2.1.2. Neuron Input and Output. In convolutional neural networks, neurons can be used in both the convolutional layer and the sampling layer. In a convolution and sampling process, only one layer of the convolutional layer and the sampling layer uses neurons, and you can freely decide which layer to use:

(1) *Neuron Input and Output of the Convolutional Layer*. In the forward propagation process of neural network, it generally includes multiple convolutional layers. Each convolutional layer takes the result of the previous convolutional layer as input [21, 22], and outputs its own output result to the next convolutional layer. There are many neurons on each convolutional layer.

The input and output process of the i-th neuron in the nth convolutional layer is as follows: this neuron takes the output of all neurons connected to the previous convolutional layer as the input and adds a deviation b. The output of the previous layer of neurons is represented by the variable x, and the input of the i-th neuron is multiplied by the weight corresponding to each neuron [23, 24]. Then, add up all the outputs, and we use the variable y to represent this activation value. The output x of this neuron is calculated by the activation function. The activation function is represented by F. The calculation process of the output value of the convolutional layer neuron is shown in

$$X_{n}^{i} = F\left(y_{n}^{i}\right) = F\left(\sum_{e=0}^{C_{n-1}} \left(W_{n}^{ie} X_{n-1}^{e} + b\right)\right). \quad (10)$$

In the convolution layer, the convolution kernel can be used to fold the feature map of the upper layer, and the output feature map can be obtained by activating the function. Each output feature map may be the result of folding the values of multiple input characteristic graphs. Each feature graph has an offset b, and the deviation of each layer is different [25, 26].

(2) *Input and Output of Neurons in Sampling Layer*. In the forward propagation process, the operation of neurons in the sampling layer is actually a down-sampling process. The number of feature graphs output by this algorithm is the same as that of the

previous convolution layer, but the size of each feature graph is reduced. The calculation formula of neuron output is

$$X_j^e = f\left(\beta_j^e \text{down}\left(X_j^{e-1}\right) + b_j^e\right). \tag{11}$$

The difference from the calculation of the convolutional layer neuron is that a downsampling function down and a multiplicative deviation β are used. Assuming that the sampling factor is n, the values of all pixels in different $n*n$ regions in the feature map are weighted and summed. Two methods of operation can be used here, the average value or the maximum value. In this way, the image is reduced to $1/n$ in both the horizontal and vertical directions. In this way, a downsampling operation process is completed, and the features of the local area of the image are collected.

2.1.3. Normalization Algorithm. The process of normalization is used to further extract features. It is divided into normalization within the same feature map and normalization within different feature maps [27–29]. First, let us talk about normalization on the same feature map. For normalization on the same feature map, we still need to perform convolution first and then correct the linearity through an activation function. The normalization is performed in a certain area through the normalization function. Therefore, we must first define a parameter to declare the size of the normalized area. The normalized calculation function is

$$f\left(u_f^{x,y}\right) = \frac{u_f^{x,y}}{\left(1 + a/N^2 \sum_{x'=\max(0,x-[N/2])}^{\min(S,x-[N/2]+N)} \sum_{y'=\max(0,y-[N/2])}^{\min(S,y-[N/2]+N)} \left(u_f^{x',y}\right)^2\right)^{\beta}}, \tag{12}$$

where $u_f^{x,y}$ represents the mapping position (x, y) of f before the activation function is normalized; S represents the pixel value of the image; and N is the size of the area used for normalization.

The normalization on different feature maps is similar to the normalization on the same feature map. The difference is that each activation unit can only be assigned by other activation units at the same location but on different feature maps. If we want to normalize a unit of the fifth feature map, then we must use the units of the third to seventh feature map. Its normalized calculation function is given as

$$f\left(u_f^{x,y}\right) = \frac{u_f^{x,y}}{\left(1 + a/N^2 \sum_{f'=\max(0,f-[N/2])}^{\min(F,x-[N/2]+N)} \left(u_{f'}^{x',y}\right)^2\right)^{\beta}}, \tag{13}$$

where F represents the number of feature planes.

2.2. Target Detection of Motion Sequence Images

2.2.1. Interframe Difference Statistics. Two images of the same background are extracted from the object motion sequence at different moments for comparison, and the result of the motion of the object under this background can be reflected through the change of its position. It is a simpler and faster method to find "difference" or subtract the gray values of the extracted two images and then it is easy to find the motion information of the object from the data obtained after subtraction.

In the obtained data value, the difference is zero, that is, the part where the gray level does not change, that is, the frame time when the object is stationary (most of the static background and a small part of the target). If the target gray scale is greater than the background, the front area is negative. The back area is positive and the other parts are zero. Extract the position of the moving object on the image from the detected part, find the movement trajectory, and further narrow the search range.

Extracting the contour of the moving object can be obtained from the difference image, the positive or negative part of the difference in the image, and then the logical sum of the sequence image is directly taken, so that the basic contour of the moving object can be extracted. Because of the influence of noise, the image of the actual moving object generally does not use a simple subtraction method but is calculated by the traditional method of interframe difference statistics.

The video sequence records the movement and change information of the video object in a certain period. The ideal video segmentation method based on moving images is to use the related information between frames for a long time to compare and judge the obtained data. Based on this idea, analyze the law of each pixel point along the time axis and select appropriate points from the entire video sequence according to the statistical law and restore the background.

The object image sequence is defined here as $I(x, y, i)$, where x, y are spatial coordinates, i is the number of frames $I = (1, 2, \ldots, N)$, and N is the total number of frames in the sequence. $I_L(x, y, i)$ is the brightness of the sequence, and the gray scale changes between adjacent frames are reflected by the video frame difference $C\ DM$:

$$C\ DM(x, y, i) = \begin{cases} d, & \text{if } d \geq T, \\ 0, & \text{if } d < T, \end{cases} \quad d = \left|I_L(x, y, i+1) - I_L(x, y, i)\right|. \tag{14}$$

T is a threshold value used as a limit value to remove noise. The coordinate position (x, y) is fixed, then $C\ DM(x, y, i)$ is a function of the number of frames i, and

what it records is the change of the pixel at the fixed position (x, y) on the time axis curve. Then, divide this curve according to whether $C\ DM(x, y, i)$is greater than zero as its dividing point and then use the part of the detected static frame to set $(S, (x, y), 1 \le j \le M)$ meaning that among them, ST_j and EN_j, respectively ,represent the start and end of S_j. Next, extract the spatial coordinates in the set (S_j) corresponding to each position (x, y), select the longest stationary segment, and record the corresponding $M(x, y)$ in the segment frame number. Finally, the recorded point $M(x, y)$ is used to fill the corresponding position in the video background. This logic can be described by

$$M(x, y) = \frac{(ST(x, y) + EN(x, y))}{2}, \tag{15}$$

$$B(x, y) = I(x, y, M(x, y)). \tag{16}$$

Among them, the starting point and ending point of the longest static segmentation correspond to $ST(x, y)$ and $EN(x, y)$, and $B(x, y)$ is the reconstructed video background. This method is based on the ideal hypothesis that moving objects will not always stand still in a certain position but will move away in the video sequence and finally reveal the background.

2.2.2. Classic Moving Target Detection Method. The optical flow method is a classic moving target detection method, which can detect moving targets and reflect the motion feature information of moving targets. Assuming that the gray value of the point Q in the image (x, y) at time t is $H(x, y, t)$, then after the interval d_t has elapsed, the horizontal and vertical motion components of this point are shown in

$$S = \frac{d_x}{d_t}, \tag{17}$$

$$C = \frac{d_y}{d_t}. \tag{18}$$

Expanded by Taylor expansion, ignoring the second-order infinitesimal, the basic constraint equation of optical flow is shown in

$$-\frac{\partial H}{\partial T} = \frac{\partial H}{\partial x} S + \frac{\partial H}{\partial y} C = \begin{bmatrix} \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} \end{bmatrix} \begin{bmatrix} S \\ C \end{bmatrix}. \tag{19}$$

The *HS* algorithm is an algorithm in the optical flow method, which introduces global smoothness into the optical flow constraint equation, and realizes the combination of two-dimensional velocity field and gray scale. The calculation formula for the deviation error of smoothness is shown in

$$E_\alpha(S, C) = \iint \left[\left(\frac{\partial S}{\partial x} \right)^2 + \left(\frac{\partial S}{\partial y} \right)^2 + \left(\frac{\partial C}{\partial x} \right)^2 + \left(\frac{\partial C}{\partial y} \right)^2 \right] \mathrm{d}x\mathrm{d}y. \tag{20}$$

Constant brightness requires that the error of the basic constraint equation of optical flow is as small as possible. The error calculation is shown in

$$E_\beta(S, C) = \iint \left[I_x S + I_y C + t \right]^2 \mathrm{d}x\mathrm{d}y. \tag{21}$$

Combining formulas (20) and (21), the optical flow in the HS algorithm should satisfy the minimum value of

$$E_\alpha(S, C) = \iint \left\{ \left(I_x S + I_y C + t \right)^2 + \mu \left[\left(\frac{\partial S}{\partial x} \right)^2 + \left(\frac{\partial S}{\partial y} \right)^2 + \left(\frac{\partial C}{\partial x} \right)^2 + \left(\frac{\partial C}{\partial y} \right)^2 \right] \right\} \mathrm{d}x\mathrm{d}y. \tag{22}$$

Among them, μ is the smoothing control parameter. When the detection image has higher resolution and less noise, the smaller the value of μ is obtained; otherwise, it should be increased. Make

$$\begin{aligned} &F\left(x, y, S, C, S_x, S_y, C_x, C_y\right)\mathrm{d}x\mathrm{d}y \\ &= \left(I_x S + I_y C + t\right)^2 + \mu \left[\left(\frac{\partial S}{\partial x} \right)^2 + \left(\frac{\partial S}{\partial y} \right)^2 + \left(\frac{\partial C}{\partial x} \right)^2 + \left(\frac{\partial C}{\partial y} \right)^2 \right]. \end{aligned} \tag{23}$$

Then, there are

$$E_{HS}(S, C) = \iint F\left(x, y, S, C, S_x, S_y, C_x, C_y\right)\mathrm{d}x\mathrm{d}y. \tag{24}$$

The inverse function of formula (24) is equivalent to solving the following equations:

$$\begin{cases} F_S - \frac{\partial}{\partial x} F_{S_x} - \frac{\partial}{\partial y} F_{S_y} = 0, \\ F_C - \frac{\partial}{\partial x} F_{C_x} - \frac{\partial}{\partial y} F_{C_y} = 0. \end{cases} \tag{25}$$

2.2.3. Adaptive Motion Detection Method. The difference method has limitations. It is only suitable for the case where the fluctuation of the background Krata is small. When the fluctuation of the background Krata of the two-frame image is large, the simple difference method cannot be used to obtain a satisfactory solution. At present, when the signal is relatively low, the background display panel and noise can be suppressed as much as possible, and adaptive motion detection methods are used to detect unstable image signals. The signal-to-clutter ratio here refers to SCNR, that is, when there is a large background clutter, the conventional

threshold segmentation method cannot separate this moving target. The adaptive motion detection method can solve this problem well, but there is a condition as the premise that the clutter background of the current image and the referenced image must be spatially related.

The figure below is a schematic diagram of the algorithm of the adaptive filter, as shown in Figure 1, clutter + noise 1, represented by $z(n) = s(n) + v_1(n)$, clutter + noise 2, represented by $x(n) = sb(n) + v_2(n)$, which is the reference input of the filter.

This method can detect moving objects adaptively and adjust the weighting coefficient, that is, the correlation between the reference image and the background of the input image, so that the output results can eliminate the influence of correlation factors on them and further compress the background clutter and correlation noise. Only by reducing the interference caused by clutter can the target be detected easily. This method is especially effective in tracking small targets.

We define a $T(n)$, which represents the actual number of instructions executed by the program in a perfectly ideal calculator. The execution time of a program is not entirely related to the amount of input, and the quality of the algorithm will also affect it, so we can regard it as a function of the amount of input n. Here, again, we can define the maximum execution time for the input n, which is the measure of time complexity. Usually, we use O to represent this execution program, where n represents the actual number of loop executions, $n0$ represents the number of predicted loop executions; $(n + l)\ (n + 2)/6 \leqq \mathrm{cn3}$, $n \geqq n0$, it means that the execution program is scientifically effective.

3. Experimental Analysis of Sports Sequence Images

3.1. Experimental Data Set. The experiment in this study uses the CIFAR-10 image data set. CIFAR-10 is an image data set containing 60,000 color pictures. The size of each picture is 32∗32. It is divided into 10 categories, and each category contains 6000 images. Recognize them by comparing different methods, and compare their recognition conditions under the same guarantee premise.

CIFAR-10 is divided into 5 training files and 1 test file. Each file contains 10,000 pictures. The test file is composed of 1000 pictures randomly selected from each category. The training file contains the remaining pictures. It is out of order, so although each training file contains 5000 pictures, some files may have more pictures in a certain category than other files.

3.2. Experimental Model Structure. The network structure of this research is roughly in accordance with the design. Take CIFAR-10 as an example. First, input 32∗32∗3 image data and pass 3∗3 convolution kernels. The number of convolution kernels is 16, and the output is 32∗32∗ 16 data, and then through $6n$ layers of convolution, because each multiscale residual learning module has two layers of convolution, there are a total of $3n$ multiscale residual learning module; after each n learning modules, we will divide the size by 2 and multiply the number of convolution kernels by 2. The downsampling operation here is realized by the convolution operation of stride = 2, and the pooling operation is not used. After the convolution of these $6n$ layers, there is a global average pooling layer, followed by a dropout operation, and finally a fully connected layer and Sotfmax. The complete network structure is shown in Figure 2, where m is the scaling parameter. We define the depth of the entire neural network as the sum of all layers with learning parameters, that is, the first 3∗3∗16 convolutional layer, $6n$ convolutional layers, and fully connected layers, totaling $6n + 2$ layers.

3.3. Experimental Data Preprocessing. Before the image data are transmitted to the network, necessary preprocessing work can significantly improve the accuracy of the network. However, in order to facilitate the comparison with the existing experimental results, we only used conventional pretreatment methods in this study.

For the training process, this study fills each picture with 0 in four directions to make it a 36∗36 picture, then randomly crops it into a 32∗32 picture and then randomly flips it left and right and uses ZCA for the picture. In the training process, there will be a shuffle process, that is, the input order of each epoch picture is random.

For the test process, this study directly input the original image data without any preprocessing operations. It will not disrupt the order.

3.4. Statistics. All data analyses in this article use SPSS19.0, statistical test uses two-sided test, significance is defined as 0.05, and $p < 0.05$ is considered significant. The statistical results are displayed as mean ± standard deviation ($x \pm \mathrm{SD}$). When the test data obey the normal distribution, the double T test is used for comparison within the group, and the independent sample T test is used for comparison between the groups. If the regular distribution is not sufficient, two independent samples and two related samples will be used for inspection.

4. Analysis of Experimental Results of Sports Sequence Images Based on Convolutional Neural Network

4.1. Analysis of Sports Sequence Images. The motion information extraction process is based on the difference image. The difference image refers to the image composed of the absolute value of the corresponding pixel gray difference between two consecutive frames in the sequence image. Ideally, through this subtraction operation, the still part of the image will be eliminated, and only those moving parts will remain. However, in actual situations, due to environmental changes in light and shade and noise, the difference image often contains some static parts. After getting the difference image, in order to facilitate subsequent calculations, the difference image needs to be binarized. Here, an adaptive threshold (threshold) technology is adopted.

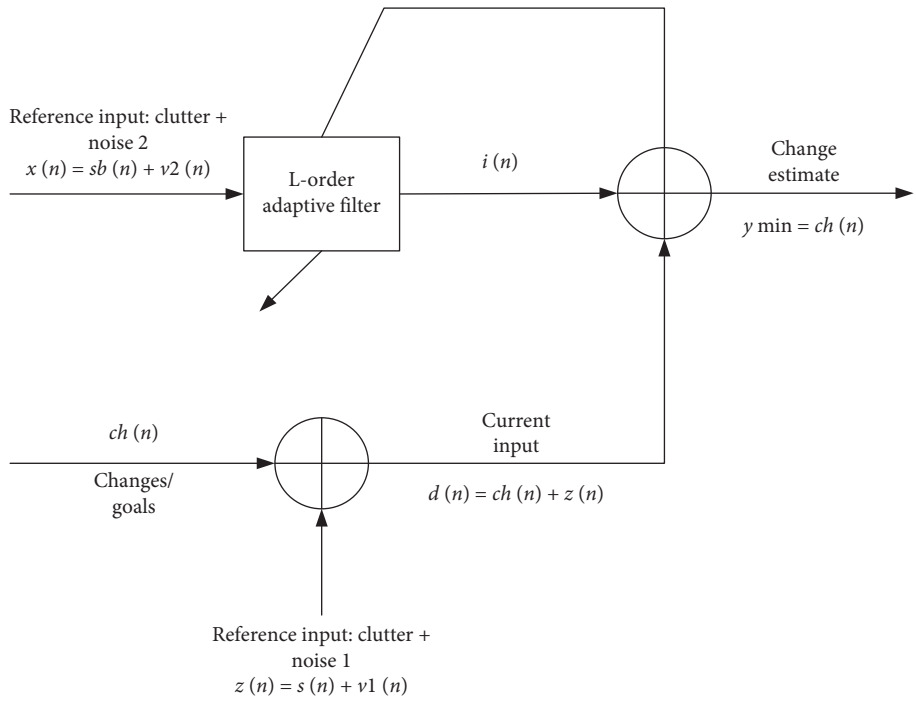

Figure 1: Schematic diagram of the adaptive algorithm.

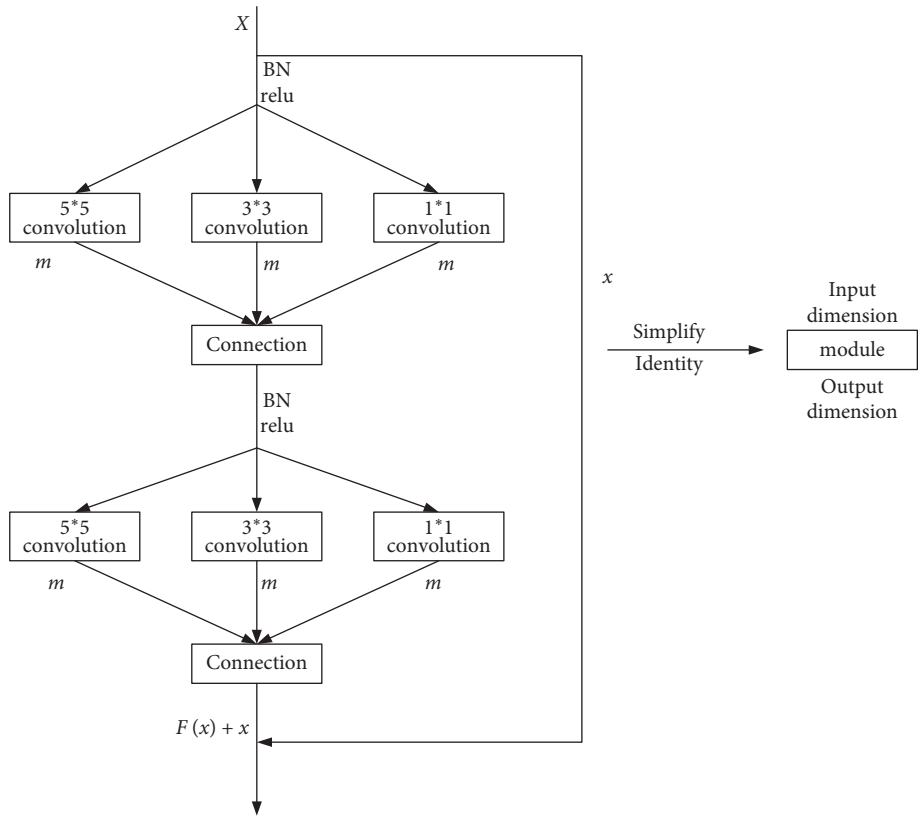

Figure 2: The complete network structure of the multiscale residual neural network.

First, make two assumptions: (1) The area of the still part of the image is larger than the area of the moving part. (2) For the images obtained with high-speed cameras, the background environment changes between two consecutive frames are very small. Obviously, these two assumptions are in line with the general situation.

As shown in Figure 3, it is an example of histogram distribution of different images. Find out the positions of several troughs in the histogram distribution curve, and the area between the troughs represents the number of pixels in different regions. According to the above assumptions (1) and (2), the segmentation points of the static region and the moving region are at the trough after the area with the largest area between the troughs. In this way, the binary threshold can be dynamically selected according to different sequence images, which can achieve better results. Thus, the difference image processed by binarization contains two parts: bright spot and black spot. The black spot represents the moving part of the image, and the bright spot represents the static part of the image.

Between two consecutive images, if the joint is still, the region in the difference image is still area, which does not contain moving pixels, that is, difference, the white point in the image; if the joint moves, the joint area in the difference image contains certain motion information, including the fixed motion pixel, that is, the black spot in the difference image. In order to judge whether the joints move in the current image, the prior knowledge of the position of each joint in the previous image should be used, that is, the coordinates of each joint in the previous frame image. The position of each joint in the first frame of the sequence image needs to be marked manually. We define the joint motion weight as S. Firstly, according to the different conditions of each joint, a template with different sizes (a specific image area) is established. According to the coordinates of each joint, the corresponding coordinates in the difference image are taken as the center. Within the range of joint template size, the number of motion pixels is searched, and the appropriate threshold is selected to judge whether the joint is moving. The calculation of joint motion weight s is shown in Figure 4.

Figure 4 shows the neighboring city within the template range with a joint coordinate as the center in the difference image. The number of motion pixels (i.e., black pixels) falling within the neighborhood range is 7, so the motion weight of the joint point $S = 7$. In the real situation and high-speed sequence images, there are only a few joints that move between two consecutive frames. Through the work of this step, the scope of the next calculation is greatly reduced.

Comparing the tracking error and time of the traditional algorithm and the algorithm in this paper, the tracking error of the traditional algorithm and the algorithm in this paper are represented by $W1$ and $W2$, and the tracking time is represented by $T1$ and $T2$.

Because the environment in which motion is always changing, if the detection method of moving targets in a dynamic background continues to use the target detection method in a static background, you will find that the binary target detection result image will contain a lot of noise, and it may be severe. The tracking target is completely covered up, thus making the tracking process fail. As shown in Figure 5, we can see that motion sequence images based on convolutional neural networks can effectively avoid such problems.

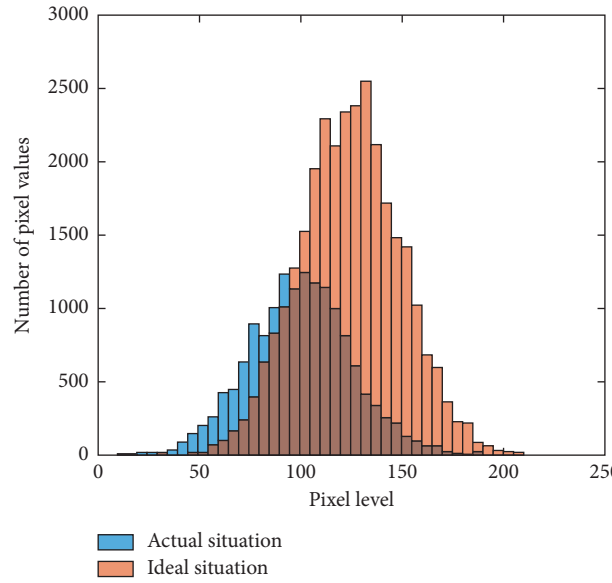

Figure 3: Histogram distribution of actual difference images.

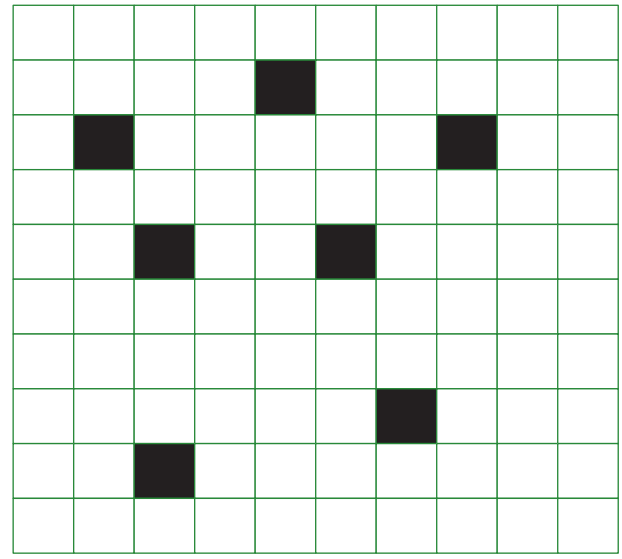

Figure 4: Calculation example of joint motion weight S.

4.2. Convolutional Neural Network Image Analysis. After weighing the training time, accuracy and proficiency of the convolutional neural network, its basic parameter settings are shown in Table 1:

This study gives the complete structure of the multiscale residual neural network, which includes $3n$ multiscale residual learning modules. Under the condition that each learning module has 2 layers of convolution, the entire network has $6n + 2$ layers, and each layer convolution also has a scaling parameter m. When each layer is composed of three different scale convolutions of $1*1$, $3*3$, and $5*5$, the width of our network is $3*m$. In order to explore the impact of different scaling parameters and depths on network performance, this study conducted multiple experiments on the CIFAR-10 data set. The results are shown in Table 2.

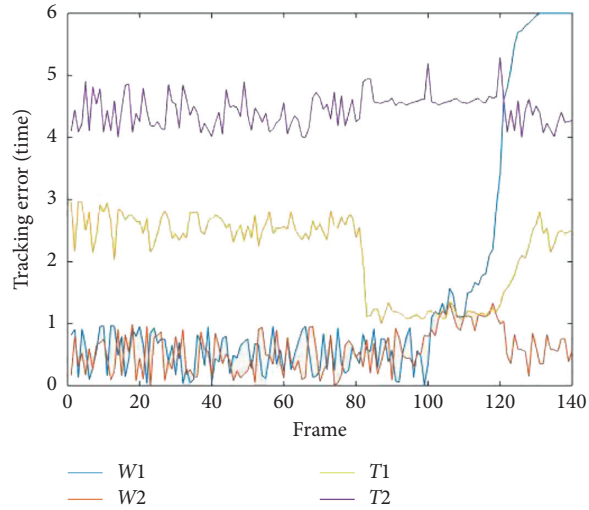

Figure 5: Comparison of tracking error and time between traditional algorithm and the proposed algorithm.

Table 1: Basic parameter settings.

Parameter	Value
Enter image size	64×64
Batch size	256
Initial learning rate	0.15
Initial learning rate decay rate	0.15
Attenuation interval	300
Dropout ratio	0.5
Weight attenuation term	[0, 0.002]
Maximum iteration steps	600

In order to verify the superiority of the improved weight optimization algorithm, this study selected the public CIFAR-10 data set as the experimental data set, and two different neural network models: 6-layer convolutional neural network and 16-layer convolutional neural network as the experimental model and parameters. The optimization algorithms Adagrad, RMSprop, and Adam are compared. For each algorithm, try several different initial learning rates {10-1, 10-2, 10-3, 10-4} and several different learning rate decay coefficients {0, 10-2, 10-3, 10–4}, finally select the pair of parameters with the highest accuracy in the test set. The convolutional neural network with several layers of convolutional layer, pooling layer, fully connected layer, and nonlinear activation function is proved to have excellent performance in real image recognition tasks. In this experiment, a 6-layer convolutional neural network is constructed, in which the first 4 layers are successively superimposed convolutional layers, and the number of filters in each layer is 32, 32, 64, 64, the first and third convolutional layers. Then, add the dropout layer with dropout rate equal to 0.25; the last two layers are fully connected layers, and the weight parameters are 1600×512, 512×10, respectively. The BatchNorm layer is added after the second fully connected layer; the activation function uses the ReLu function, and the pooling layer uses MaxPooling method. The final experimental comparison results of each algorithm are shown in Figure 6.

The change trend of accuracy of different algorithms on the test set is shown in Figure 7.

As can be seen from Figure 6, although the improved algorithm in this paper lags behind the Adam algorithm in the initial convergence speed, the convergence speed gradually surpasses the Adam algorithm after the 12th round and converges to a lower loss function value. The accuracy rate change curve of the test set in Figure 7 shows that the improved algorithm in this paper finally converges to the highest accuracy rate, indicating that the improved algorithm has better parameter optimization capabilities.

4.3. Actual Application Results of the Proposed Algorithm

4.3.1. Application in Basketball Sequence Images. This algorithm is applied to the classification and goal tracking of basketball sequence images. The algorithm is used to count the number of goals and scores of team A and B in the 5 basketball games and compare them with the actual number of goals and scores.

As shown in Table 3, after statistics on the goals and scores of the two teams in 5 games, it is found that the algorithm in this paper still has some errors in actual application. In one game, the number of missed or over-remembered goals occurred, and 2 games were missed or overremembered. The specific comparison of the number of goals is as follows:

As shown in Figure 8, in the third game, the algorithm of this paper counts A team scored 25 times and actually scored 27 times, counted B team scored 30 times, and actually scored 29 times. The scores are as follows:

As shown in Figure 9, in the first game, the statistical team A scored 87 points, and the actual score was 89 points; the statistical team B score was consistent with the actual score, which was 81 points. In the third game, team A scored 49 points and the actual score was 52 points; team B scored 59 points and the actual score was 57 points.

Table 2: Test set error rate under different network structures.

Model number	Scaling parameters m	Network depth	Parameter	Test set error (%)
1	1	74 ($n = 12$)	13.4M	4.80
2	1	98 ($n = 16$)	17.7M	4.90
3	1	122 ($n = 20$)	22.2M	4.95
4	2	50 ($n = 5$)	34.9M	4.20
5	3	32 ($n = 6$)	48.2M	4.19
6	3	38 ($n = 4$)	58.3M	4.21
7	4	26 ($n = 3$)	67.5M	3.99

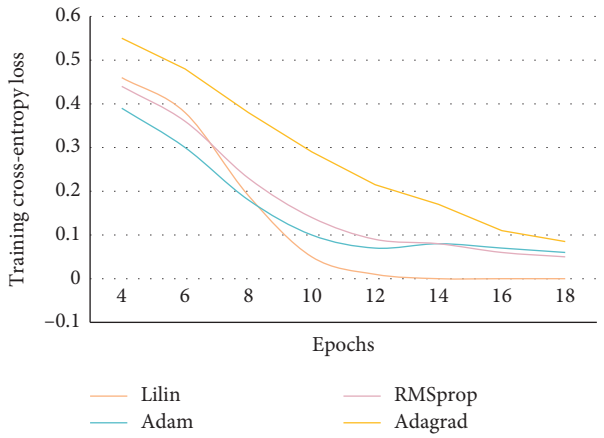

Figure 6: The change trend of the loss function value of different algorithms on the training set.

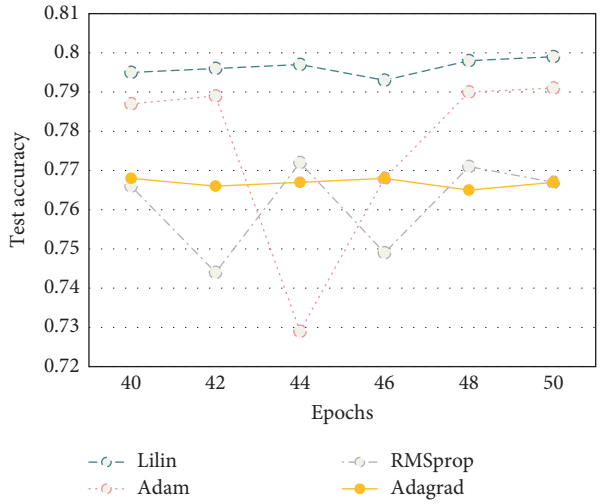

Figure 7: Trends of accuracy of different algorithms on the test set.

4.3.2. Application in Sprint Motion Sequence Images. The algorithm of this paper is applied to the classification and target tracking of sprint motion sequence images, and the situation of a certain class of college students ($n = 42$) in the 100-m sprint physical test is tracked throughout the course. Among them, there are 18 boys and 6 in groups, which are 1–3 groups; 24 girls

Table 3: Number of goals and score statistics.

Sessions	Algorithm in this paper	Actual	Difference
Number of goals (A1)	36	36	0
Number of goals (B1)	32	32	0
Number of goals (A2)	21	21	0
Number of goals (B2)	24	24	0
Number of goals (A3)	25	27	2
Number of goals (B3)	30	29	−1
Number of goals (A4)	36	36	0
Number of goals (B4)	30	30	0
Number of goals (A5)	19	19	0
Number of goals (B5)	15	15	0
Total number of goals	268	269	1
Score (A1)	87	89	2
Score (B1)	81	81	0
Score (A2)	38	38	0
Score (B2)	43	43	0
Score (A3)	49	52	3
Score (B3)	59	57	−2
Score (A4)	70	70	0
Score (B4)	61	61	0
Score (A5)	45	45	0
Score (B5)	36	36	0
Total score	569	572	3

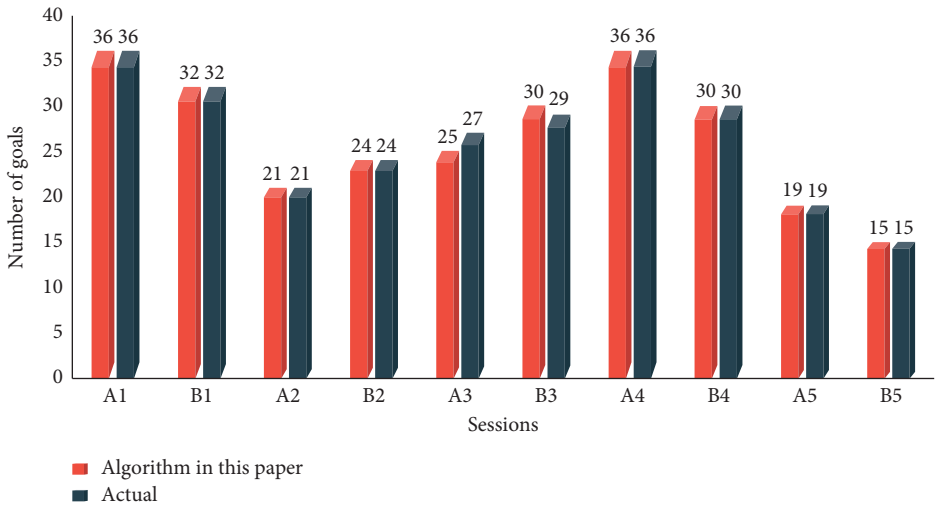

Figure 8: Comparison of the number of goals scored.

and 6 in groups, which are 4–7 groups. The average score of each group is calculated and compared with the actual results.

As shown in Tables 4 and 5, even if the algorithm statistics are the same as the actual highest and lowest scores, their average scores are not consistent. There are errors in the average scores of 3 out of 7 groups. The details of the average grade are as follows:

As shown in Figure 10, the algorithm statistics average score of the first group is 12.9 s, and the actual average score is 12.6s. The algorithmic statistical average score of the fourth group is 16.5 s, and the actual average score is 16.4 s. The algorithmic statistical average score of the sixth group is 16.2 s, and the actual average score is 16.7 s.

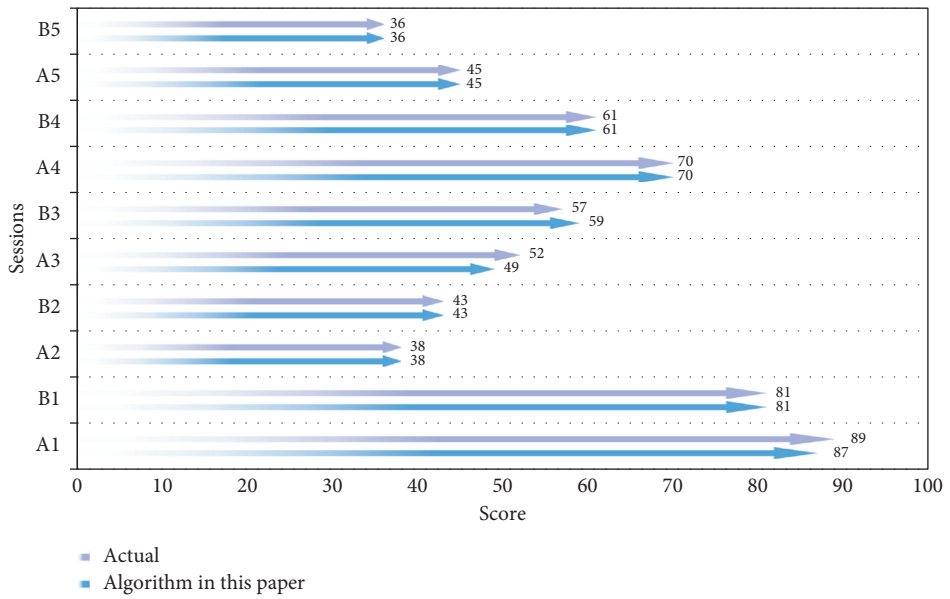

Figure 9: Comparison of score statistics.

Table 4: Algorithm to calculate the sprint performance of each group of students.

Group	Average score (s)	Highest score (s)	Minimum score (s)
1	12.9	12.5	15.1
2	13.1	12.6	15
3	12.8	12.6	14.8
4	16.5	15.7	18.1
5	16.4	15.5	18.5
6	16.2	15.5	17.9
7	16.8	15.9	19

Table 5: Actual sprint results of each group of students' performance.

Group	Average score (s)	Highest score (s)	Minimum score (s)
1	12.6	12.5	15.1
2	13.1	12.6	15
3	12.8	12.6	14.8
4	16.4	15.5	18.1
5	16.4	15.5	18.5
6	16.7	15.5	17.9
7	16.8	15.9	19

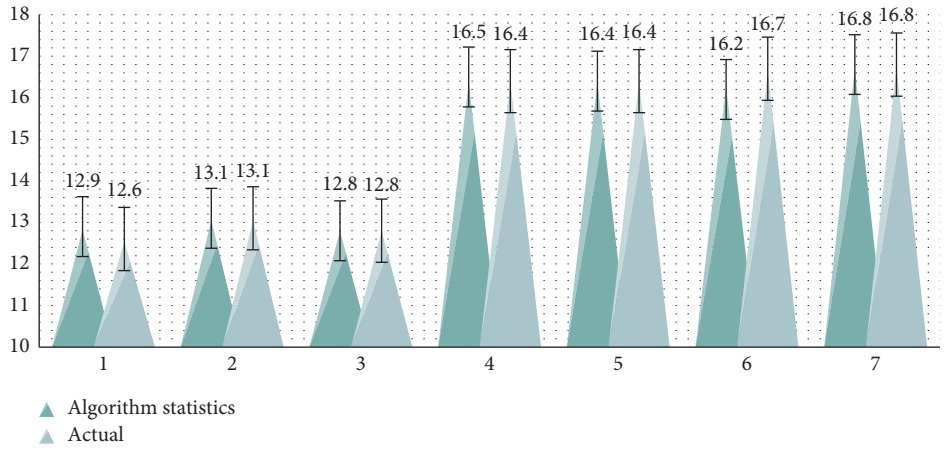

Figure 10: Comparison of average scores for each group.

5. Conclusion

In this paper, firstly, aiming at the problem of sports sequence image analysis in complex scenes, two improvements are made to the VGG convolutional neural network model with excellent performance. Firstly, aiming at the problems of too many parameters of the original model and the limitation of image input size, the network structure is improved to further reduce the over fitting of the model and improve the flexibility of the model; secondly, the performance of the VGG convolutional neural network is improved. Aiming at the problem that the classification accuracy of the original model is not ideal in complex scenes, the element of target detection is added to get a hybrid model with double loss function. One of the optimization objectives of the loss function is the location of the bounding box coordinate points and the length and width of the bounding box of the output target object, and the other is the correct classification of the output target object. Because the two objective functions share the weight parameters of convolution layer for feature extraction, the feature mapping biased to the coordinate region of the target object is extracted from the optimized model.

In order to further improve the optimization speed and ability of weight parameter optimization algorithm, this paper proposes an improved optimization algorithm based on Adam algorithm: in the aspect of adaptive adjustment of learning rate, an improved adaptive adjustment method of learning rate is proposed, the main content is adding a new feedback mechanism; in the aspect of learning rate annealing, a periodic annealing method is proposed, The main content is to optimize the annealing method of the improved algorithm, and integrate a periodic annealing method. The experimental results show that the improved method is superior to other optimization algorithms.

This article also has some shortcomings. The method used mainly tests the images of people in motion. But in reality, the scene often contains both people and other moving targets such as cars. The behavior of the target person may also suddenly take place in large movements. People are overcrowded in places, such as shopping malls, waiting halls, and people for a long time. The case of depth occlusion. Therefore, similar problems need further research and analysis.

Data Availability

The data underlying the results presented in the study are included within the manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This work was supported by Regular Project of Shaanxi Sports Bureau in 2020 (No. 2020105).

References

[1] X. Yang and S. Jiang, "Research on theory and method for facial expression recognition system based on dynamic image sequence," *Open Automation & Control Systems Journal*, vol. 7, no. 1, pp. 569–579, 2015.

[2] A. Guermazi and F. W. Roemer, "Compositional MRI assessment of cartilage: what is it and what is its potential for sports medicine?" *British Journal of Sports Medicine*, vol. 50, no. 15, pp. 896-897, 2016.

[3] S. He, R. W. H. Lau, W. Liu, Z. Huang, and Q. Yang, "SuperCNN: a superpixelwise convolutional neural network for salient object detection," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 330–344, 2015.

[4] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[5] G. Xiao, R. Wang, C. Zhang, and A. Ni, "Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks," *Multimedia Tools and Applications*, 2020.

[6] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua, "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network," *Knowledge-Based Systems*, vol. 132, no. 15, pp. 62–71, 2017.

[7] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Auto-context convolutional neural network (Auto-Net) for brain extraction in magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 36, no. 11, 2017.

[8] C. Yuan, X. Li, Q. M. J. Wu et al., "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Computers, Materials and Continua*, vol. 53, no. 4, pp. 357–371, 2017.

[9] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.

[10] X. Li, Y. Wang, and G. Liu, "Structured medical pathology data hiding information association mining algorithm based on optimized convolutional neural network," *IEEE Access*, vol. 8, no. 1, pp. 1443–1452, 2020.

[11] A. Sevastopolsky, "Optic disc and cup segmentation methods for glaucoma detection with modification of U-net convolutional neural network," *Pattern Recognition and Image Analysis*, vol. 27, no. 3, pp. 618–624, 2017.

[12] F. C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 99, pp. 4392–4400, 2018.

[13] Z. Zhao and A. Kumar, "Accurate periocular recognition under less constrained environment using semantics-assisted convolutional neural network," *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 5, 2017.

[14] T. Hirasawa, K. Aoyama, T. Tanimoto et al., "Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images," *Gastric Cancer Official Journal of the International Gastric Cancer Association & the Japanese Gastric Cancer Association*, vol. 87, no. 1, pp. 1–8, 2018.

[15] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1207–1216, 2016.

[16] P. Chen, L. Zheng, X. Wang et al., "Moving target detection using colocated MIMO radar on multiple distributed moving platforms," *IEEE Transactions on Signal Processing*, vol. 65, no. 99, 2017.

[17] X. Chen, J. Guan, Y. He et al., "High-resolution sparse representation and its applications in radar moving target detection," *Journal of Radars*, vol. 6, no. 3, pp. 239–251, 2017.

[18] L. Deng and H. Zhu, "Moving point target detection based on clutter suppression using spatiotemporal local increment coding," *Electronics Letters*, vol. 51, no. 8, pp. 625-626, 2015.

[19] G. Gao, G. Shi, L. Yang, and S. Zhou, "Moving target detection based on the spreading characteristics of SAR interferograms in the magnitude-phase plane," *Remote Sensing*, vol. 7, no. 2, pp. 1836–1854, 2015.

[20] S. Jie, C. Fu-qing, Z. Cai-sheng, and H. You, "Experimental results of maritime moving target detection based on passive bistatic radar using non-cooperative radar illuminators," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6763–6766, 2019.

[21] E. Jaya and B. T. Krishna, "Fuzzy-based MTD," *Data Technologies and Applications*, vol. 54, no. 1, pp. 66–84, 2020.

[22] Q. Gong and C. Wang, "Moving target detection algorithm based on sparse recovery and sample selection for airborne radar," *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 40, no. 5, pp. 1012–1017, 2018.

[23] L. Hai, S. Wenyu, L. Weijian et al., "Moving target detection with limited training data based on the subspace orthogonal projection," *IET Radar Sonar & Navigation*, vol. 12, no. 7, pp. 679–684, 2018.

[24] Y. Zhao, H. Li, S. Wan et al., "Knowledge-aided convolutional neural network for small organ segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1363–1373, 2019.

[25] Y. Ma, H. Hong, and X. Zhu, "Multiple moving-target indication for urban sensing using change detection-based compressive sensing," *IEEE Geoence and Remote Sensing Letters*, vol. 18, no. 99, 2020.

[26] Z. Li, F. Santi, D. Pastina et al., "A multi-frame fractional fourier transform technique for moving target detection with space-based passive radar," *IET Radar Sonar & Navigation*, vol. 11, no. 5, pp. 822–828, 2016.

[27] G. Zhaoming, J. Yi, and B. Shihua, "Detection probability for moving ground target of normal distribution using an imaging satellite," *Chinese Journal of Electronics*, vol. 27, no. 6, pp. 1309–1315, 2018.

[28] M. Elhoseny and K. Shankar, "Optimal bilateral filter and convolutional neural network based denoising method of medical image measurements," *Measurement*, vol. 143, pp. 125–135, 2019.

[29] Z. M. Zhu and X. B. Wang, "The research of ultra wide-band in searching and rescuing rader micro-moving target detection methods," *Computing Techniques for Geophysical & Geochemical Exploration*, vol. 37, no. 2, pp. 141–144, 2015.