Hindawi
Scientific Programming
Volume 2022, Article ID 9436736, 11 pages
https://doi.org/10.1155/2022/9436736

Research Article

The Motor Action Analysis Based on Deep Learning

TianYu Zhang

Sanjiang University, Ministry of Sports, Nanjing 210000, China

Correspondence should be addressed to TianYu Zhang; 2009040141@st.btbu.edu.cn

Received 7 December 2021; Revised 27 December 2021; Accepted 11 January 2022; Published 10 March 2022

Academic Editor: Hangjun Che

For the slow speed and low accuracy of slow motor action recognition methods, this study proposes a motor action analysis method based on the CNN network and the softmax classification model. First, in order to obtain motor action feature information, by using static spatial features of BN-inception based on CNN network extracted actions and high-dimensional features of 3D ConvNet, then based on softmax classifier structure and realizing taxonomic recognition of the motor actions. Finally, through the decision-layer fusion and time semantic continuity optimization strategy, the motion action recognition accuracy is further improved and the more efficient motion action classification recognition is realized. The results show that the proposed method can complete the motor action analysis and achieve the classification recognition accuracy to 83.11%, which has certain practical value.

1. Related Work

Movement action analysis is an important branch of computer vision, which also involves data mining, image processing, and other content, and is widely used in sports, music playing, and many other scenes. Due to the complex patterns of movement action and the big differences in movement rules of different individuals, the movement action recognition analysis is somewhat challenging and has attracted the keen attention of relevant researchers. At present, motion action analysis mainly focuses on motion detection and recognition and has achieved remarkable research results. For example, Hua-xin Zhang et al. realized the estimation of human posture by capturing 3D motion [1]. In addition, Xiaoqiang Li et al. applied the convolutional neural network to action recognition. The results show that the action recognition results of a convolutional neural network with the dual-attention mechanism are comparable to the recognition results of the latest algorithm [2]. Haohua Zhao et al. extracted intraframe feature vectors by deep network training to form a multimode feature matrix. The matrix is input into CNN to achieve feature classification. The results show that the proposed method has better performance than the existing LSTM in video action recognition [3]. Ran Cui et al. analyzed the motion by constructing skeletal joints and static and dynamic features. The prediction of motion is realized through motion recognition [4]. Manikandaprabu et al. detected the ROI of the human body using the combination of background subtraction and frame subtraction [5]. Then the CAMShift algorithm is adopted for recognition. The results show that this method has good precision and has great advantages compared with the most advanced algorithms. It can be seen from the above studies that convolutional neural networks are widely used in action recognition, among which the CNN attracts more attention due to its unique characteristics.

Despite the great progress in motor motion analysis, its overall performance still needs to be improved, mainly due to the blurred boundary of motor motion, which increases the difficulty of the study. For the difficulties, this study applies powerful deep learning capabilities, based on the CNN network and the softmax classifier, and proposes a deep learning-based motion action analysis method.

2. Basic Methods

2.1. Network Profile. The CNN network is a representative algorithm of deep learning, which is commonly used in image processing, video image recognition, and other fields,

with the characteristics of simple structure and strong expansion performance. Its basic module includes a convolution layer, a pooling layer, and a full connection layer, as shown in Figure 1. The convolution layer is responsible for extracting the local features of the input image to obtain different feature maps; the pooling layer reduces the dimension of the extracted features of the convolution layer to retain important information while reducing the risk of overfitting due to nonessential information. Common pooling layer settings include average pooling and maximum pooling; the full connection layer plays a classification role in the network and enables sample data classification by mapping the learned feature data to the space of sample markers [6–11].

In recent years, with the deepening of deep learning research, a huge breakthrough in CNN network structure has been made. In terms of spatial feature extraction, the network continuously deepens, forms the inception structure module, as shown in Figure 2, which greatly reduces the quantities of network parameters, realizes the multiscale processing fusion of images, and obtains a better feature representation [12–15].

In terms of spatiotemporal feature extraction, a 3D ConvNet network emerged, acquiring spatiotemporal features by performing both convolutional and pooling operations in time and space simultaneously, further improving the model performance.

2.2. Softmax Model Introduction. The softmax model is a multiclassifier based on the logistic regression model that can handle multiclassification problems. In the softmax model, for a given input x, the hypothetical function hθ(x) was used to estimate the probability value of each category j, p ($y=j$|x), i.e., estimating the probability of each classification result of x. Suppose the k-dimensional vector output by the function is the probability of these estimated k values. The hθ(x) form is as follows:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p\left(y^{(i)} = 1|x^{(i)};\theta\right) \\ p\left(y^{(i)} = 2|x^{(i)};\theta\right) \\ \vdots \\ p\left(y^{(i)} = k|x^{(i)};\theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}. \quad (1)$$

In the formula, $\theta_1, \theta_2, \ldots, \theta_k \in \Re^{n+1}$ represents model parameters, $1/\sum_{j=1}^{k} e^{\theta_j^T x^{(x)}}$ is to normalize the probability distribution so that the sum of all probabilities is one.

Thus, the probability that softmax classifies x into category j can be expressed as [16–18]

$$p\left(y^{(i)} = j|x^i;\theta\right) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}. \quad (2)$$

Motor action analysis is a multitaxonomic recognition process. According to the above analysis, in order to better analyze motor actions, this paper first used the feature extraction method to acquire motor representation based on the CNN network and then identified by the softmax classifier to realize the analysis of motor actions.

3. Motor Motion Analysis Method Based on Deep Learning

3.1. Characteristic Extraction Based on CNN Network. The analysis of motor actions includes the appearance features and action context information of the data. In order to obtain well robust action characteristics, this study, based on the CNN network, represents the motion action appearance features and motion features by extracting the low-dimensional static features and high-dimensional spatial and temporal features of the data, respectively, to represent the motion action features, as shown in Figure 3.

3.1.1. Static Spatial Characteristic Extraction. In this paper, the BN-inception network with high accuracy and efficiency extracts the static spatial features of motion action, whose network structure is shown in Table 1. Specific extraction steps are as follows [19–22]:

Step 1: pretreatment for image cutting, motion action images, and image level flipping to obtain a matrix that meets the BN-Inception network input

Step 2: tacking the input matrix through a pretrained BN-inception model with feature extraction and calculating the feature average of each dimension of different image parts according to equation (3)

$$d_j = \frac{1}{M}\sum_{i=1}^{,M} fea_{i,j}, \quad j = \{1, 2, \ldots, 10\}, j = \{1, 2, \ldots 1024\}. \quad (3)$$

Step 3: obtaining final feature representation of a single-frame image, as in formula (4)

$$\text{Static}_{\text{fea}} = (d_1, d_2, \ldots, d_{1024}). \quad (4)$$

Decreeing $f = \text{Static}_{\text{fea}}$, a characteristic representation of a sample of motion action data is a two-D matrix $F = \{f_1, f_2, \ldots, f_N\}$ of N x D. In it, N represents the total number of motion action video segment frames, fn represents the single-frame image feature, and D represents the feature dimension size. And so forth, all the static spatial features of the motor movement can be obtained.

3.1.2. Dynamic Spatiotemporal Feature Extraction. In this paper, 3D ConvNet high-dimensional spatial features and the network structure are shown in Figure 4. The specific extraction method is as follows:

Step 1: A multiscale frame sequence is entered and divides the video into different scale segments according to the set window size

Step 2: the spatiotemporal feature representation of each segmentation timing segment fc6 layer is extracted by network forward propagation, such as follows:

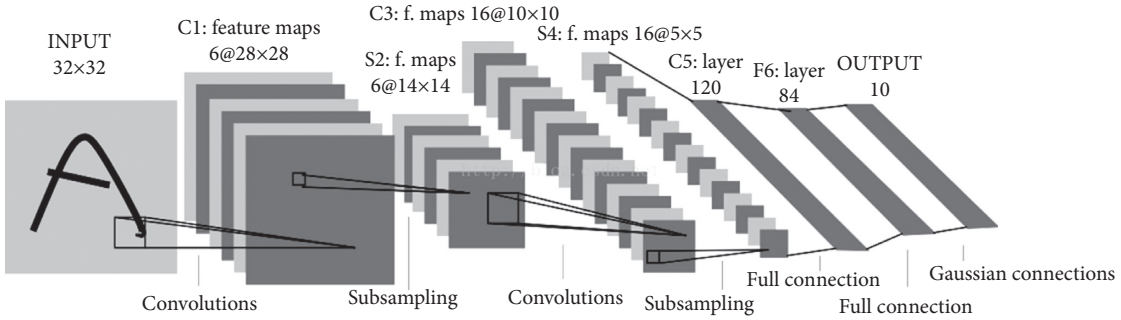

Figure 1: Standard CNN network structure.

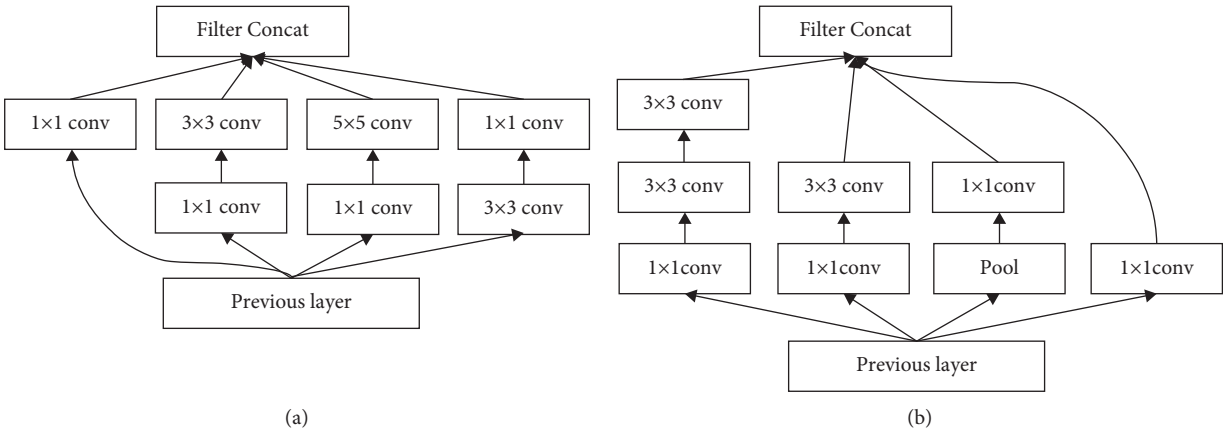

Figure 2: Structural representation of inception. (a) Inception v1 module. (b) Inception v2 module.

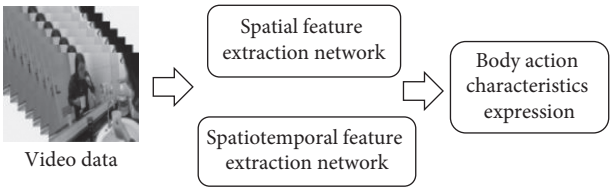

Figure 3: Structured flowchart of deep feature extraction.

Table 1: BN-inception network structure.

Type	Kernel size/step	Output size
Convolution	7 × 7/2	112 × 112 × 64
Max pool	3 × 3/2	56 × 56 × 64
Convolution	3 × 3/1	56 × 56 × 192
Max pool (3a)	2 × 3/2	28 × 28 × 192
Inception (3b)		28 × 28 × 2S6
Inception (3a)		28 × 28 × 320
Inception (3c)	stride2	14 × 14 × 576
Inception (3a)		14 × 14 × 576
Inception (4a)		14 × 14 × 576
Inception (4b)		14 × 14 × 608
Inception (4c)		14 × 14 × 60×
Inception (5a)	stride2	7 × 7 × 1056
Inception (5b)		7 × 7 × 1024
Inception (5c)		7 × 7 × 1024
Avg pool	7 × 7/1	1 × 1 × 1024

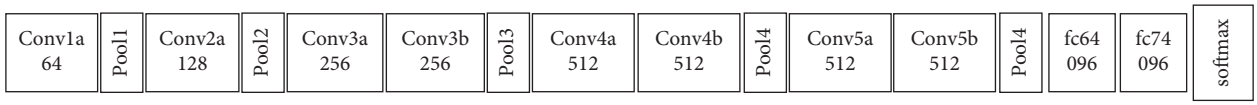

Figure 4: 3D ConvNet network structure block diagram.

$$\text{Dynamic}_{\text{fea}} = (d_1, d_2, \ldots, d_{4096}). \tag{5}$$

Decreed $f = \text{Dynamic}_{\text{fea}}$, for a sample of motion action video data with a total frame number as N, if the time overlap is 50%, the extracted action feature representation $F = (f_1, f_2, \ldots, f_K)$ is a $K \times D$ 2D matrix. Where $K = \lfloor (N - (w_i/2))/(w_i/2) \rfloor$, fk represents the input fragment features, and D represents the dimension size. With the above operation, the spatiotemporal features of all motion action video samples are extracted.

3.2. Classification Identification of Motion Actions Based on the Softmax Model

3.2.1. Model Structure Construction. Based on the above feature extraction, the softmax model structure was designed as shown in Figure 5 in this study. This classification network includes three fully connected layers for selecting parameters, one dropout layer to prevent overfitting, and finally, connecting the softmax loss. During training, the parameters were optimized by using small-batch gradient descent [23–25].

Considering that CNN network-based features include low-dimensional static features extracted by BN-inception and high-dimensional spatiotemporal features extracted from C3D, to improve the classification effect, different dimensions were trained separately in the study. To set the number of full connected-layer neurons of the low-dimensional feature softmax classification network for fc1 = 512, fc2 = 256, fc3 = 6, and fc1 = 1024, fc2 = 512, and fc3 = 6, while the high-dimensional feature softmax classification network for fc1 = 1024, fc2 = 512, fc3 = 6.

3.2.2. Model Training and Testing. Specific procedures of training and testing of the above softmax model are as follows:

Step 1: The feature matrix of the training sample data is built. Assuming training sample QTY is M, the feature matrix of sample i (i = 1,2,3, …, M) is $F_i = N$ x D, N represents the number of training sample frames, and D represents the size of the feature dimension extracted per frame. The total number of M samples can be represented as

$$F = \sum_{i=1}^{M} F_i. \tag{6}$$

To train the soft model by the network structure in Figure 5, the number of fc3 output neurons is the same as in the categorical category C and the output vector $X = \{x_i\}$, $j = 1, 2, \ldots, C$. Therefore, the corresponding marker probability y_j for the output value x_j obtained by the softmax function can be expressed as [26]

$$y_j = \text{softmax}(x)_j = \frac{\exp(x_j)}{\sum_{j=1}^{C} \exp(x_j)}. \tag{7}$$

Step 2: to minimize the loss, a cross-entropy loss function was used, as shown in equation (8) to minimize the loss during training.

$$H_y(y) = -\sum_j y_j \log(y_j). \tag{8}$$

In the abovementioned formula, y_j represents the score distribution of classification by softmax; and y_j represents the target true value. Thus, the C category final-loss-value loss is the average of the cross-entropy loss for each category, as in the following formula[27]:

$$\text{Loss} = \text{mean}(H_y(y)). \tag{9}$$

Step 3: after optimizing the training parameters to speed up the model training, the model parameters were optimized by using M-BGD. During optimization, weight optimization is as in formula (10)–(14) [28].

$$\text{errors} = y_j - \text{softmax}(x_j). \tag{10}$$

$$\Delta w = \alpha \times x_j \times \text{errors}, \tag{11}$$

$$\Delta \beta = \alpha \times \text{errors}, \tag{12}$$

$$w = w - \Delta w, \tag{13}$$

$$\beta = \beta - \Delta \beta, \tag{14}$$

where errors represent the weight error, α represents the learning rate, w represents the weight, and β represents the deviation.

Step 4: model testing is performed. The best softmax classification model obtained by training is selected to classify and identify the test dataset, and the corresponding action category of the maximum classification score in C categories output from the following layer is selected as the classification result of the test data, and it is expressed as follows:

$$\text{class} = \text{index}(\max\{y_1, y_2, \ldots, y_c\}). \tag{15}$$

3.3. Decision-Making Layer-Based Fusion. Considering the diversity, complexity, and ambiguity among motor

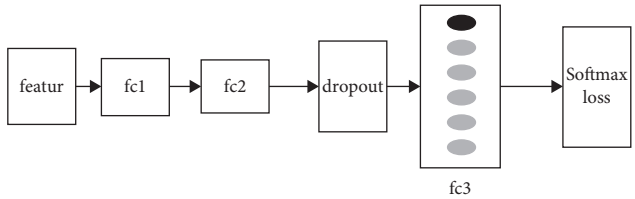

Figure 5: Structure diagram of the classification network based on softmax.

movements and the different key movements, different movement performance modes need to be mixed together to improve the classification and recognition accuracy of motor movements. Currently, common fusion methods include feature-based and decision-based fusion. Since feature-based fusion stitched and fused the features, it may lead to mutual interference among features and learning efficiency. In contrast, the fusion method based on the decision-layer only needs to determine action categories based on different classification confidence sizes, which is efficient and simple. Therefore, this paper fuses the classification results in a decision-layer fusion-based manner.

The fusion structure based on the decision-layer fusion mode is shown in Figure 6. Assuming the number of classified motor action categories is C, for individual test data X, the classification result is where $\text{result}(X) = \{s_1, s_2, \ldots, s_C\}$, in it, s_i represents the classification score of category i, $i \in \{1, 2, \ldots, C\}$, the N road classification identification results can be summarized according to formula (16).

$$\text{result}(X_i') = \frac{1}{N}\sum_{n=1}^{N}\text{result}\left(X_n^i\right), \quad n \in \{1, 2, \ldots, N\}. \tag{16}$$

Then to sum and average the data to obtain the final classification results

$$\text{result}(X)' = \{s_1', s_2', \ldots, s_3'\}. \tag{17}$$

In the formula, X_n^i represents the classification score of sample X in the classifier n, and X_i' is the classification score of X is category i after fusion. The classification result based on $\text{result}(X)'$ maximum is the final classification category of sample X, as in the following formula:

$$\text{label} = \max\{\text{result}(X_i')\}, \quad i \in \{1, 2, \ldots C\}. \tag{18}$$

3.4. Time-Based Semantic Continuity Optimization. Movement actions have a certain time sequence, so there are a large number of redundant and incomplete trivial fragments during sequential action detection. To further improve the detection performance of the method according to temporal semantic continuity, this study proposes an optimization strategy based on the characteristics of motion action.

First, to model the motion action time sequence semantics and time sliding window classification at different scales, initial detection results are taken. Defined all detection results of a motor action as X, $seg(c_i, w_k)$ indicated a

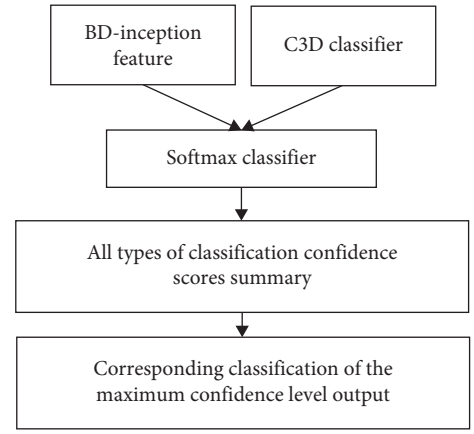

Figure 6: Fusion structure block diagram.

category of c_i in X, and a set of tests with a sliding window size of w_k can be represented as

$$X = \{\text{seg}(c_i, w_k)\}, \tag{19}$$

$$\text{seg}(c_i, w_k) = \{(s_n, e_n, g_n)\}_{n=1}^{N_{i,k}}. \tag{20}$$

In the formula, C represents the total number of categorical categories; K represents the total number of sliding windows; $N_{i,k}$ is the number of action time segments detected in c_i and w_k; s_I, e_I are the start and end times of detected action segments; and gn represents the classification score.

Then to calculate the classification score, the temporal overlap values of different time periods, as in equations (21) and (22), and compare them with the set threshold.

$$|g_I - g_s| < \theta, \tag{21}$$

$$\text{IOU}(p_l, p_s) \geq U \times \min(T_l, T_s) l, s \in \{1, 2, \ldots N_{i,k}\}, \ \vdots \ l \neq s. \tag{22}$$

In the formula, $P = \text{seg}(c_i, w_k)$ represents the detection results of the same category c_i and the same scale w_k, pl = (sl, el, gl), and ps = (ss, es, gs) both are one detection fragment of P; $|g_l - g_s|$, $\text{IOU}(p_l, p_s)$ represent the score difference and time overlap value, respectively; $T_l = e_l - s_l$, $T_s = e_s - s_s$ the execution time of two actions, and θ, U the set threshold.

The two action segments were integrated if the two action segment classification scores were less than the set threshold and the time overlap was greater than the threshold.

Considering that the motion action obtained by the above operation is synthesized from multiple incomplete fragments, which partially destroys the spatiotemporal structure of the action, it also needs to conduct classification detection. This study uses a 3D convolutional neural network with good classification performance for reclassification. Furthermore, to ensure more accurate classification results and reduce the classification impact of sliding windows on motor movements, it statistically calculated the weight scores of different sliding windows for different

categories, further adjusted the classification confidence scores, and trained classification models by softmax classifier and overlap loss function.

Finally, to reduce redundancy detection in the presence of sliding windows at different scales, they were processed by nonmaximum inhibition to bring the final results close to the start and end of the motor action.

4. Simulation Experiment

4.1. Data Source and Preprocessing. The project dataset and the Thoumos14 publicdata set were used for this experiment. The project dataset contains 72 video action segments of six action categories, including brushing, mouthwash, and cleaning, with the characteristics of complex background, variable perspective, and an obvious difference in action execution speed. Its specific description is shown in Table 2. The Thoumos14 public dataset includes 2,755 clips in 20 sports action categories, with a total of 212 test videos annotated with timing. Because there were two mislabels of "270" and "1496" in this dataset, the remaining 210 annotated timing videos were selected for this experiment.

4.2. Parameter Settings

4.2.1. BN-Inception Network Parameter Settings. The BN-inception network parameters of this experiment are set as in Table 3.

4.2.2. 3D ConvNet Network Parameter Settings. In this experiment, the 3D ConvNet network parameters were set as follows: the convolutional kernel size was $3 \times 3 \times 3$, the step size was $1 \times 1 \times 1$, the size of the first pooling layer was $1 \times 2 \times 2$, and the size of the remaining pooling layers to $2 \times 2 \times 2$ with maximum pooling.

4.2.3. Softmax Classifier Parameter Selection. The M-BGD optimized softmax classifier parameters were used for this experiment. First, the batch-size size of the softmax classifier was selected. The average accuracy change curve on the project dataset test set under the same number of iterations when different batch-size values are taken in Figure 7. It is known from Figure 7(a) that when batch-size = 64, the test had the highest average accuracy, hence batch-size = 64 is set. Second, the number of iterations is selected. The effect of the different number of iterations on the identification results during training is shown in Figure 7(b). As Figure 7(b) shows that the highest identification result was achieved when the number of iterations was 20000, so the number of iterations was set at 20000.

During the training session, the loss change curves are shown in Figure 8. This figure shows that the loss values gradually decrease and tend to 0 during training.

4.2.4. Thresholding Selection Based on Temporal Semantic Continuity. The choice of score difference optimized threshold θ has a certain impact on the integration speed of the detection window. If θ value is too large, it will easily lead to excessive integration of the detection window; if θ value is too small, it will lead to the integration time fragments cannot being merged, and there are still too many incomplete time fragments. Therefore, this experiment determined reasonable values by analyzing the influence of different θ values on mAP. Figure 9 shows the mAP taking different values at different temporal overlap thresholds on the Thoumos14 public dataset. And the highest mAP value is when $\theta = 0.5\text{e}{-3}$. Therefore, this experiment was set $\theta = 0.5{-3}$.

Considering the time continuity of motion movements and the variability of sliding windows at different scales, the time overlap threshold was set for $U = 2/3$.

4.3. Results and Analyses

4.3.1. Softmax Classified Network Performance Analysis. To validate the performance of the proposed softmax classification network, this study was validated on the project dataset and compared with the SVM classification network, and the results are presented in Table 4. According to the table, compared with the SVM classifier, the softmax classifier has a better effect, achieves a classification identification accuracy of 78.52%, and improves by 12.22%. Moreover, with the same classification recognition accuracy, the proposed softmax network in this study has a shorter training time and looks about 10 times shorter than the SVM classifier. This shows that the softmax classification network proposed in this study performs better and is more conducive to motion action analysis.

4.3.2. Fusion Result Analysis Based on the Decision Layer. To verify the effectiveness of the decision-layer-based fusion method proposed in this study, the study was validated on the project dataset and compared with the prefusion classification identification results, results are shown in Table 5. According to the table, the average classification recognition accuracy reached 79.89%, an improvement of 1.38% compared with before the fusion method, indicating that the fusion method has some effectiveness.

4.3.3. Validation Based on the Temporal Semantic Continuity Optimization Method. To further verify the effectiveness of this temporal semantic-based continuity optimization method, the study was validated on the project dataset and compared the identification results of partial test samples before and after optimization, as shown in Table 6 and Table 7. According to the table, this proposed method in this study can effectively improve the recognition accuracy from 79.89% to 83.11%.

In the test dataset, the mAP values at different time overlap thresholds are shown in Table 6. According to the table, when $\alpha = 0.5$, the average detection accuracy of the present study is 60.2%.

4.3.4. Classification Identification Results Analysis. To verify the effectiveness of the proposed method, this study visualized the method classification identification results in

Table 2: Project dataset description.

Difficulties	Description
Background noise	Different individual clothing changes, relationships with the environment, etc
Action nonstandardized	There are interruptions in the process of action execution, disorderly order, property management action, etc
Speed difference of action execution	Differences in speed and performance time during action execution occur due to the behavioral habits of different individuals
Different perspectives	Camera location, different angles, etc
Shelter	Cameras and their own occlusion problems
Lens movement	There are varying degrees of lens movement during the dataset recording process

Table 3: BN-inception network parameter settings.

type	Depth	#1 × 1	#1 × 1 reduce	#3 × 3	#3 × 3 reduce	Double #3 × 3	Pool + proj
Convolution	1						
Max pool	0						
Convolution	1	64	192				
Max pool	0						
Inception (3a)	3	64	64	64	64	96	Avg + 32
Inception (3b)	3	64	64	96	64	96	Avg + 64
Inception (3c)	3	0	128	160	64	96	Max + pass
Inception (4a)	3	224	64	96	96	128	Avg + 128
Inception (4b)	3	192	96	128	96	128	Avg + 128
Inception (4c)	3	160	128	160	128	160	Avg + 128
Inception (4d)	3	96	128	192	160	192	Avg + 128
Inception (4e)	3	0	128	192	192	256	Max + pass
Inception (5a)	3	352	192	320	160	224	Avg + 128
Inception (5b)	3	352	192	320	192	224	Max + 128
Avg pool	0						

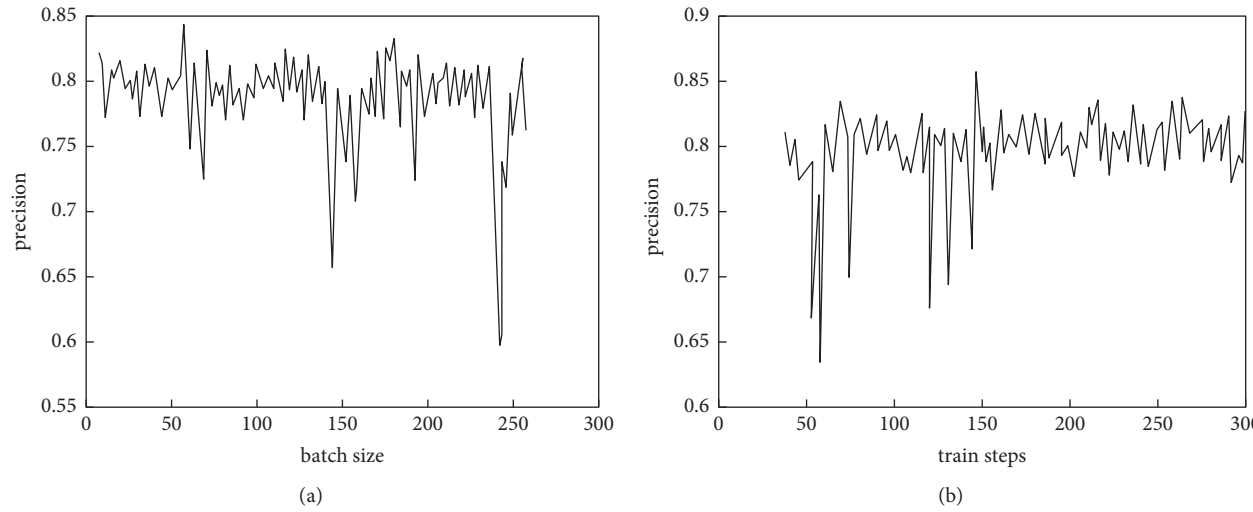

Figure 7: Selection of training parameters. (a) Effect on the identification results. (b) Effect of the number of training iterations on the identification results.

Figure 10. Figure 10(a) is the input video stream identification result, where the abscissa is the video frame, and the ordinate is the identification accuracy with the highest classification score for each frame. Figure 10(b) is the action performed at different time periods in the video stream, the abscissa represents the video stream, and the ordinate is the action category. Since the project dataset used for the experiment includes six action categories, each color in the figure corresponds to one action category, so there are six colors in Figure 10. According to the figure, the proposed method has the highest accuracy of the actions in the process. The recognition accuracy of different actions decreased due to the vague beginning and end of the operation type and time location. In Figure 10(b), green presents for

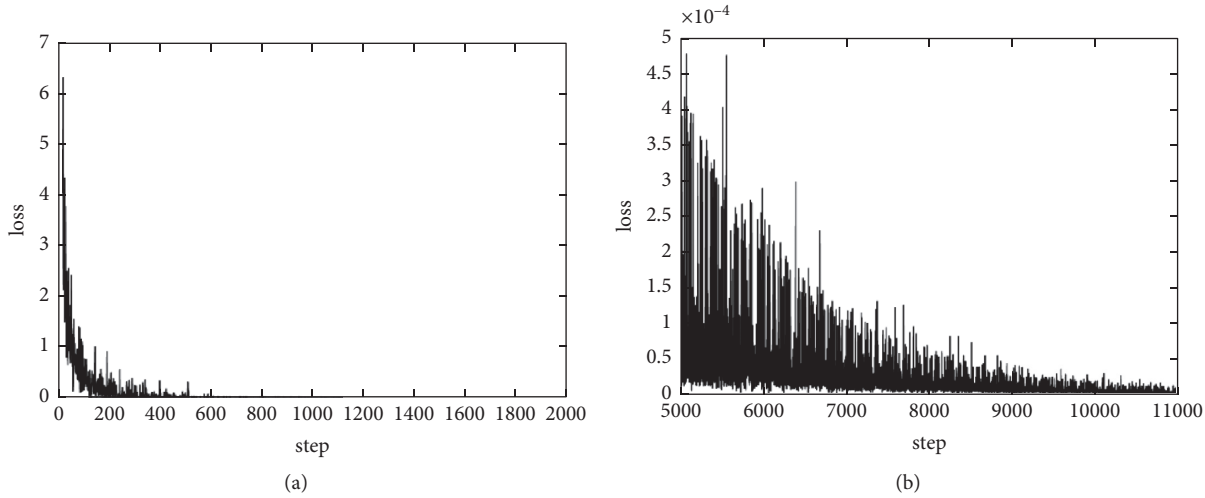

Figure 8: Loss curves of different iteration times. (a) The loss change with 0–2,000 iterations. (b) The loss change with 50000–11,000 iterations.

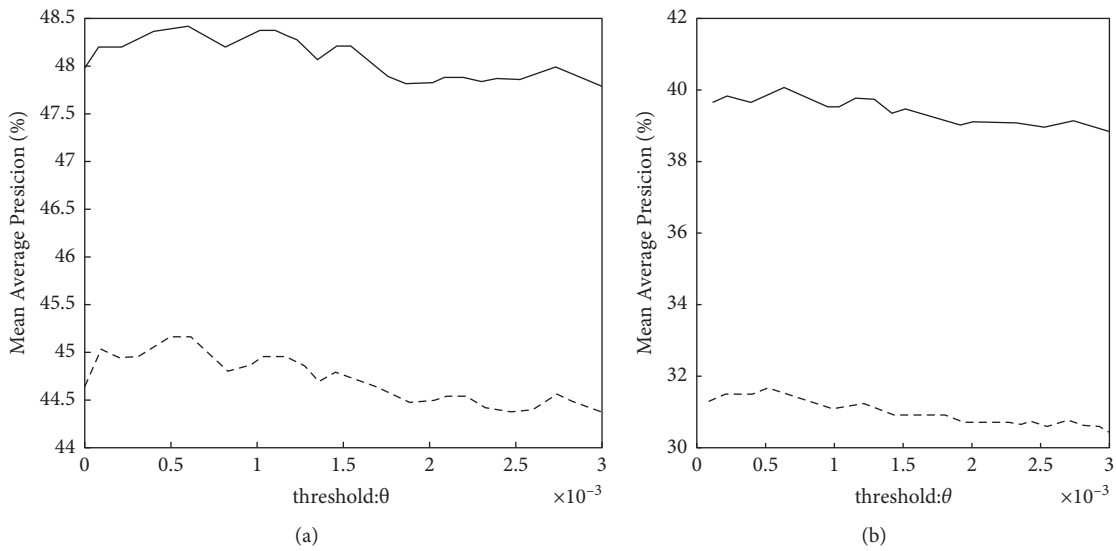

Figure 9: Different thresholds θ impact on mAP. (a) θ changing curves with mAP when A is 0.1,0.2. (b) θ changing curves with mAP when A is 0.3,0.4.

Table 4: Comparison of classification and recognition results of different classifiers.

Features test videos	Softmax	SVM
Pl	76.84	58.73
P2	80.33	59.51
P3	51.25	72.07
P4	86.39	84.24
P5	76.29	33.33
P6	943 6	94.21
P7	93.34	61.82
P8	71.69	72.82
P9	82.80	58.42
P10	71.88	67.91
AVG	78.52	66.30

Table 5: Fusion experimental results.

classifier Features test video	Softmax classifier		Class score confusion
	BN-inception features (%)	C3D features (%)	
Pl	76.84	58.73	75.91
P2	80.33	59.51	81.63
P3	51.25	72.0	63.26
P4	86.39	84.24	86.13
P5	76.29	33.33	74.29
P6	94.36	94.21	94.43
P7	93.34	61.82	94.52
P8	71.69	72.82	76.17
P9	82.80	58.42	84.41
P10	71.88	67.91	68.23
AVG	78.51	66.30	79.89

Table 6: Comparison of results before and after optimization.

Methods test videos	Class score confusion (%)	Action continuity and temporal integrated (%)
Pl	75.91	81.18
P2	81.63	8139
P3	63.26	76.97
P4	86.13	83.81
P5	74.2 V	72.02
P6	94.43	94.62
P7	94.52	95.24
P8	76.17	80.46
P9	84.41	89.78
P10	68.23	75.61
AVG	79.89	83.11

Table 7: Comparison of map values at different time overlapping thresholds.

a	0.5	0.6	0.7	0.7	0.8	0.9
mAP (%)	60.2	45.1	35.3	35.3	27.1	14.3

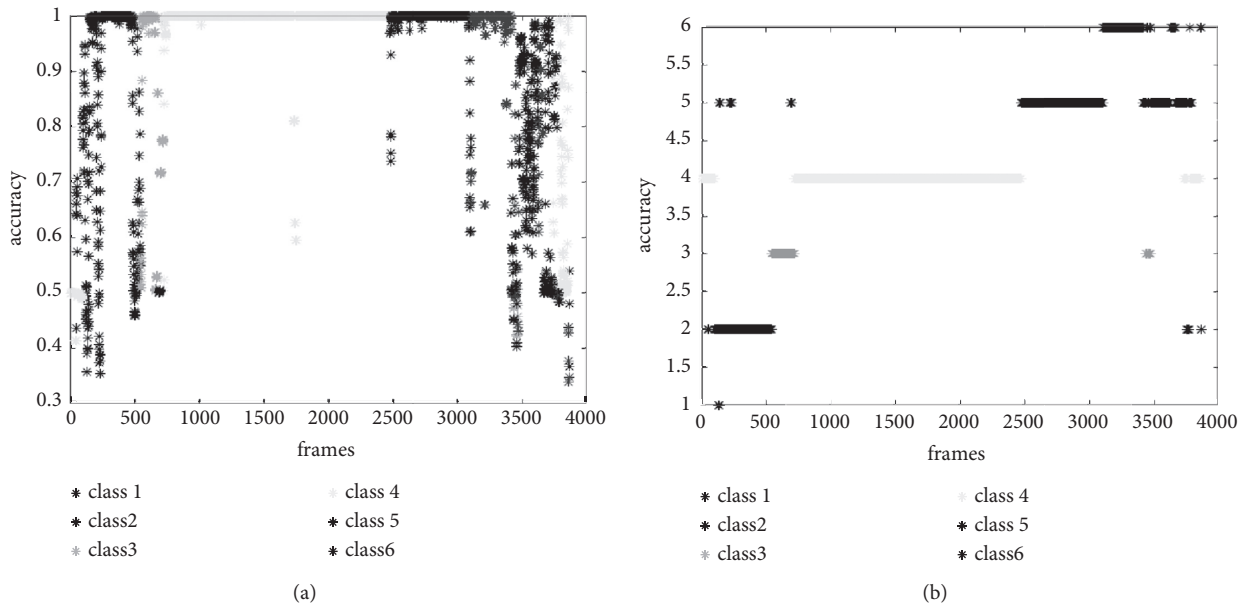

Figure 10: Visualization of classification recognition results. (a) Optimal visualization results of single frames in a video stream. (b) Classification results in the action timing flow.

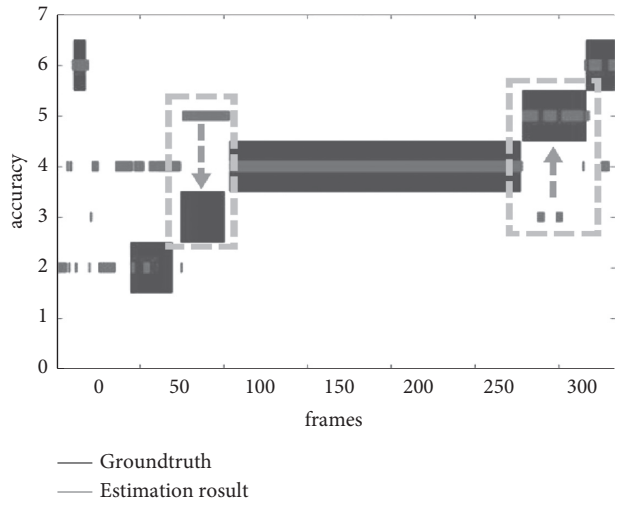

Figure 11: Visualization of classification recognition results.

the third action category. The proposed method presents the detection results (545,687), (697,672), and (3440,3468) frames, consistent with actual conditions.

To further analyze the classification and recognition effects of the similarity action, a study performed taxonomic identification of a test sample, and the results were shown in Figure 11. In the figure, abscissa represents video streams, the ordinate represents action categories, blue represents true values, red represents predictive values, and green boxes represent misclassification caused by similarity actions. According to the figure, considering the action time timing structure is conducive to the analysis of motion movements.

5. Conclusion

In summary, using the deep-learning-based motor action analysis method proposed in this study, static spatial features of the motion action are extracted by using BN-inception and high-dimensional spatiotemporal features of motor movements are extracted by 3D ConvNet. A characteristic representation containing the spatiotemporal information of the motor movements is obtained. By using the softmax classifier and integrating the extracted features with the fusion based on decision layers, the accuracy of motion action classification and recognition was improved, so that the average motion action classification and recognition accuracy reached 79.89%. Through the time-based semantic continuity optimization strategy, the recognition accuracy of motion movements was further improved, and the average motion action classification recognition accuracy reached to 83.11%, realizing the efficient recognition of motion actions. However, there are still some deficiencies in this study, mainly manifested in feature extraction. The BN-inception network and 3D ConvNet network used in the study trained the model in advance for the public dataset and did not fine-tune the model structure according to the research content, so its robustness needs to be further improved.

Data Availability

The data used in this experiment are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding this work.

References

[1] X. Li, M. Xie, Y. Zhang, G. Ding, and W. Tong, "Dual attention convolutional network for action recognition," *IET Image Processing*, vol. 14, no. 6, pp. 1059–1065, 2020.

[2] H.-X. Zhang and L. Su, "An improved approach for human motion simulation and sports analysis based on dynamic image analysis[J]," *International Journal of Frontiers in Sociology*, vol. 0, no. 4, p. 2, 0, 2020.

[3] H. Zhao, W. Xue, X. Li, Z. Gu, L. Niu, and L. Zhang, "Multi-mode neural network for human action recognition," *IET Computer Vision*, vol. 14, no. 8, pp. 587–596, 2020.

[4] R. Cui, A. Zhu, J. Wu, and G. Hua, "Skeleton-based attention-aware spatial-temporal model for action detection and recognition," *IET Computer Vision*, vol. 14, no. 5, pp. 177–184, 2020.

[5] N. Manikandaprabu and S. Vijayachitra, "Moving human target detection and tracking in video frames[J]," *Studies in Informatics and Control*, vol. 30, no. 1, pp. 119–129, 2021.

[6] M. Khan, Mustaqeem, A. Ullah et al., "Human action recognition using attention based LSTM network with dilated CNN features[J]," *Future Generation Computer Systems*, vol. 125, pp. 820–830, 2021.

[7] A. V. Anant, S. C. Gupta, D. Kumar, and Savita, "Human action recognition using CNN-SVM model[J]," *Advances in Science and Technology*, vol. 6258, pp. 282–290, 2021.

[8] A. K. S. Kushwaha and R. Khurana, "Fusing dynamic images and depth motion maps for action recognition in surveillance systems," *International Journal of Sensors, Wireless Communications & Control*, vol. 11, no. 1, pp. 107–113, 2021.

[9] N. S. Russel and A. Selvaraj, "Fusion of spatial and dynamic CNN streams for action recognition[J]," *Multimedia Systems*, vol. 27, no. 5, pp. 1–16, 2021.

[10] S. K. Park, J. H. Chung, T. K. Kang, and M. T. Lim, "Binary dense sift flow based two stream CNN for human action recognition[J]," *Multimedia Tools and Applications*, vol. 80, no. 28-29, pp. 35697–35720, 2021.

[11] D. zheng, H. Li, H. Li, and S. Yin, "Action recognition based on the modified twostream CNN," *International Journal of Mathematics and Soft Computing*, vol. 6, no. 6, pp. 15–23, 2020.

[12] L. Xiao, T. Lan, D. Xu, W. Gao, and C. Li, "A simplified CNNs visual perception learning network algorithm for foods recognition," *Computers & Electrical Engineering*, vol. 92, p. 107152, 2021.

[13] G. Li and C. Li, "Learning skeleton information for human action analysis using Kinect," *Signal Processing: Image Communication*, vol. 84, p. 115814, 2020.

[14] Y. Ji, Y. Yang, F. Shen, H. T. Shen, and X. Li, "A survey of human action analysis in HRI applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2114–2128, 2020.

[15] Q. Wu, A. Zhu, R. Cui et al., "Pose-Guided Inflated 3D ConvNet for action recognition in videos," *Signal Processing: Image Communication*, vol. 91, p. 116098, 2021.

[16] W. Li, N. Xu, G. Liu, L. Zhao, and X. Fang, "Segments-based 3D ConvNet for action recognition," *Journal of Physics: Conference Series*, vol. 1621, no. 1, p. 012042, 2020.

[17] E. Barnefske and H. Sternberg, "PCCT: pcct: a point cloud classification tool to create 3D training data to adjust and develop 3D convnet," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W16, pp. 35–40, 2019.

[18] M. Lkhagvadorj, K. H. Ryu, O.-E. Namsrai, and N. Theera-Umpon, "A partially interpretable Adaptive softmax regression for credit scoring[J]," *Applied Sciences*, vol. 11, no. 7, p. 3227, 2021.

[19] J. Luo, W. Shi, N. Lu et al., "Improving the performance of multisubject motor imagery-based BCIs using twin cascaded softmax CNNs," *Journal of Neural Engineering*, vol. 18, no. 3, p. 036024, 2021.

[20] D. M. Vo, D. M. Nguyen, and S.-W. Lee, "Deep softmax collaborative representation for robust degraded face recognition," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104052, 2021.

[21] S. S. S. Palakodati, V. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning[J]," *Revue d'Intelligence Artificielle*, vol. 34, no. 5, pp. 617–622, 2020.

[22] R. Deepa, H. Rajaguru, and C. Ganesh Babu, "Analysis on wavelet feature and softmax discriminant classifier for the detection of epilepsy," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012036, 2021.

[23] F. Gao, B. Li, L. Chen, Z. Shang, X. Wei, and C. He, "A softmax classifier for high-precision classification of ultrasonic similar signals," *Ultrasonics*, vol. 112, p. 106344, 2021.

[24] C. V. R. Reddy, K. K. Kishore, U. S. Reddy, and M. Suneetha, "Person identification system using feature level fusion of multi-biometrics," in *Proceedings of the 2016 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–6, Chennai, India, 15-17 Dec. 2016.

[25] V. Chirra, S. ReddyUyyala, and V. KishoreKolli, "Deep CNN: a machine learning approach for driver drowsiness detection based on eye state," *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 461–466, 2019.

[26] L. M. R. Azizah, S. F. Umayah, S. Riyadi, and N. A. Utama, "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," in *Proceedings of the 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 242–246, Penang, Malaysia, 24-26 Nov. 2017.

[27] A. Wu, J. Zhu, and T. Ren, "Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network," *Computers & Electrical Engineering*, vol. 81, p. 106454, 2020.

[28] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.