Retraction Notice

The Editor-in-Chief and the publisher have retracted this article, which was submitted as part of a guest-edited special section. An investigation uncovered evidence of systematic manipulation of the publication process, including compromised peer review. The Editor and publisher no longer have confidence in the results and conclusions of the article.

XY, XZ, WL, XL, and PS did not agree with the retraction.

Cross-spectral human behavior recognition based on deep convolutional networks for global temporal representation

Xiaomo Yu[a,b,*] **Xiaomeng Zhou**[a] **Wenjing Li**[a,b] **Xinquan Liu**[a] **and Peihua Song**[a]

[a]Nanning Normal University, Department of Logistics Management and Engineering, Nanning, China

[b]Nanning Normal University, Guangxi Key Lab of Human-Machine Interaction and Intelligent Decision, Nanning, China

Abstract. The purpose of behavior recognition is to recognize the actions of the human body in action. It plays a great role in surveillance, video recommendation, and human-computer interaction with video. With the rise of neural networks, behavior recognition has also continued to develop and progress and has reached a relatively advanced level. However, behavior recognition is still insufficient in recognizing complex human movements and recognizing videos in different bands. To solve this problem, this paper establishes a convolutional neural network (CNN) cross-spectral human behavior recognition algorithm based on global time domain representation. It adopts the method of time-domain feature extraction, construction of optimized convolutional neural network layers, and global time-domain cross-spectrum construction. It also uses videos from the unified compliance framework (UCF)-sports and UCF-11 datasets for experiments. Experiments show that the algorithm achieves an average accuracy of 90% in the behavior recognition of UCF-sports. It still maintains an average accuracy rate of >90% in the more complex behavior recognition of UCF-11, and the highest accuracy rate is 93%. © *2022 SPIE and IS&T* [DOI: 10.1117/1.JEI.32.1.011209]

Keywords: temporal representation; deep convolutional network; cross spectrum; behavior recognition.

Paper 220367SS received Apr. 15, 2022; accepted for publication Jul. 8, 2022; published online Jul. 26, 2022.

1 Introduction

In the field of artificial intelligence, human behavior recognition is a field of great development significance, especially in the related aspects of human recognition and prediction in video recordings. In behavior recognition, artificial intelligence can recognize and analyze human behaviors to achieve information interaction. Therefore, it abandons the human-computer interaction method of keyboard and mouse. Human action recognition analysis is actually to output human action language as a kind of information. It includes gesture language, facial expression language, head language, etc. Therefore, it is very important to recognize and understand these languages to complete the output of information.

Convolutional neural networks were not originally used for action recognition, but for image classification. But because of its excellent visual processing power, it can play a great role in behavior recognition. Convolutional neural network is a multilayer neural network structure. Neurons are only connected between layers and not within themselves. The operation of the neuron is also nonlinear activation after the inner product of the input and the weight vector. The network finally outputs a score vector. It defines the loss function and uses gradient descent for optimization.

The innovation of this paper is that an efficient behavior recognition method is proposed and verified. Based on global temporal feature extraction, it introduces a convolutional neural

*Address all correspondence to Xiaomo Yu, yuxiaomo@nnnu.edu.cn

1017-9909/2022/$28.00 © 2022 SPIE and IS&T

network as a better visual processing tool. This thus enables cross-spectral human behavior recognition. It can achieve high efficiency and high accuracy even in the case of many frequency bands.

2 Related Work

The number of categories for behavior recognition is growing rapidly. And it is getting harder and harder to label enough training data and to learn regular models across all classes. Because the mapping between the video spatiotemporal features of actions and the semantic space is more complex and harder to learn. Xu et al.[1] built a mapping between visual features and semantic descriptors for each action category, which allowed new categories to be identified without any visual training data. Li et al.[2] proposed a global manifold edge learning method for data feature extraction or dimensionality reduction, which was called local linear representation manifold edge (LLRMM). It can then reconstruct any node in the two graphs using the minimum local linear representation technique. This can be deduced from the graph scatter between manifolds and the scatter within the manifold graph, which was a novel global model of manifolds. Xie et al.[3] proposed a robust and trainable spectral difference mapping method based on convolutional networks. This method performed residual learning in an end-to-end manner and can be used to remove noise in HSI while preserving spectral profiles for human action recognition. Zhao et al.[4] used a Kalman filter and smoother. After a comparative evaluation of different convolutional architectures, he proposed an efficient deep convolutional neural network to classify 2D time-domain ECG data. Xiao et al.[5] proposed new deep network architecture. It was used to remove streak noise from a single weather satellite infrared cloud image. In the proposed framework, residual learning was used to directly reduce the mapping range from input to output, thereby speeding up the training process and improving the destriping performance. Wen et al.[6] proposed a deep learning model with some convolutional filters to learn hierarchical features from data. He compared CNN models with popular machine learning algorithms: AdaBoost, random forest (RF), and support vector machines (SVM). It was found that his classification model had a significant improvement in speed compared with the traditional model. Yang et al.[7] proposed a localization segmentation framework, where localization was performed by a deep regression model based on convolutional neural networks. They also used the Dice loss function in the training process of the segmentation model to study its impact on the segmentation performance, which provided great reference value to later researchers.

3 Specific Method for Human Behavior Recognition Based on Global Temporal Feature Deep Convolutional Network Time Series is a Very Important Class of Features

Such features are not based on manual mining but are produced by machine learning models. It differs from many traditional time series-related statistical features. More importantly, it tends to give a huge boost to the model. Researchers first introduced the concept of temporal template in the field of human action recognition and proposed motion energy image (MEI) and motion history image (MHI) methods to describe actions.[8] There is a difference between global time domain features and local features. It does not need to capture local details but recognizes them based on the overall scene. Therefore, it is more efficient and more accurate in human behavior recognition. Next, the method introduced in this section integrates a convolutional neural network with time-domain representation and cross-spectral recognition, which has more functions in method features and can be well used for human behavior recognition in this paper.

3.1 *Time Domain Feature Extraction*

The time domain feature is a very common method in behavior recognition. It specifically refers to the time-related features in the process of sequence changes over time.[9] The global time domain properties are all properties based on total time. It uses n to represent the size of a

time window (i.e., the number of data rows in the window), and i to represent the i'th data row. The formula is as follows:

$$\text{mean} = \frac{1}{n}\sum_{i=1}^{\text{n}} a_i. \tag{1}$$

In the mean calculation type, the function available in MATLAB is mean and the function available in Python is numpy.mean. In the standard deviation calculation formula, the function available in MATLAB is std, and the function available in Python is std

$$\text{std} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \text{mean})^2}. \tag{2}$$

This function is a widely used time domain feature. Usually, in a dataset, the number that occurs most frequently is called the dataset function. For example, the function of 1, 2, 3, 3, and 4 is 3. However, if there are two or more numbers with the most impressions, then those numbers are the mode of operation for this dataset. Also, if all data are shown the same number of times, the dataset has no features. For the convenience of calculation, if there are multiple operation methods, we take the average as the only method. The formula for calculating the maximum value in the window is as follows:

$$\max = \max(a_i), i \in \{1,2,\ldots,n\}. \tag{3}$$

Conversely, the formula for calculating the minimum value in a window is as follows:

$$\min = \min(a_i), i \in \{1,2,\ldots,n\}. \tag{4}$$

The difference between max and min within a window is its range

$$\text{range} = |\max - \min|. \tag{5}$$

The number of cross-average points is the number of data that exceeds the average point in a window. Where $(\cdot)$ is the indicator function, which takes the value 1 when the condition in the parentheses is true, and 0 otherwise

$$\text{above\_mean} = \sum_{i=1}^{n}(a_i \geq \text{mean}). \tag{6}$$

The correlation coefficient refers to the correlation between two vectors (such as the readings of the x and y axes). It is generally used to identify movements in one direction and is not very commonly used. The calculation formula is

$$P_{x,y} = \frac{\text{cov}(x,y)}{\delta_x \delta_y}. \tag{7}$$

Among them, $\text{cov}(x, y)$ represents the Pearson correlation coefficient of x and y, and $\delta_x\delta_y$ represents the standard deviation of x and y, respectively. Generally speaking, this feature is more obvious in the static state and the moving state. Where t represents the size of a time window

$$\text{SMA} = \frac{1}{t}\left(\int_0^t |x(t)|\text{d}t + \int_0^t |y(t)|\text{d}t + \int_0^t |z(t)|\text{d}t\right). \tag{8}$$

3.2 *Building and Optimizing Convolutional Neural Network (CNN) Layers*

A convolutional neural network (CNN) is a feedforward neural network. The pattern of connections between its neurons is inspired by the animal visual cortex.[10]

In a typical CNN architecture, convolutional layers are connected together in a cascade. Each convolutional layer is followed by a rectified linear unit layer followed by a pooling layer. It is then followed by one or more convolutional layers, followed by another pooling layer. It ends with one or more fully connected layers. This depends a lot on the type of problem to be solved, so the network structure can be very deep. The output of each convolutional layer is a set of objects called feature maps, which are generated by a kernel filter. It can then use the feature map to define new inputs to the next layer. Each neuron in a CNN network produces an output, which is followed by an activation threshold. This threshold is proportional to the input, not a fixed value, as shown in Fig. 1.

Pooling layers are usually placed after convolutional layers. It divides the pooling layer into subregions and then divides the convolutional regions. To understand this more clearly, it is necessary to refer to Fig. 2. It selects a single representation value. It utilizes max pooling or average pooling techniques to reduce the computation time of subsequent layers. This way the CNN can be seen as a feature extractor.

In this way, the robustness of the feature in its spatial location is also improved. More specifically, as the feature map is passed through the grayscale image as an image attribute, it will become more and more in the process of passing through the network. But as more and more feature maps are added, it usually gets deeper and deeper.

To reduce the huge weight of data and calculation amount of C3D, it has made optimization of the common method. The usual approach is to decompose the 3D convolution into a 2D spatial convolution ($S: 3 \times 3 \times 1$) and a 1D temporal convolution ($T: 1 \times 1 \times 3$). It then links the two convolutions in series or parallel. It combines the characteristics of a residual network, which forms a pseudo-3D (P3D) network structure. It combines the three improved structures together, and the effect is shown in Fig. 3.

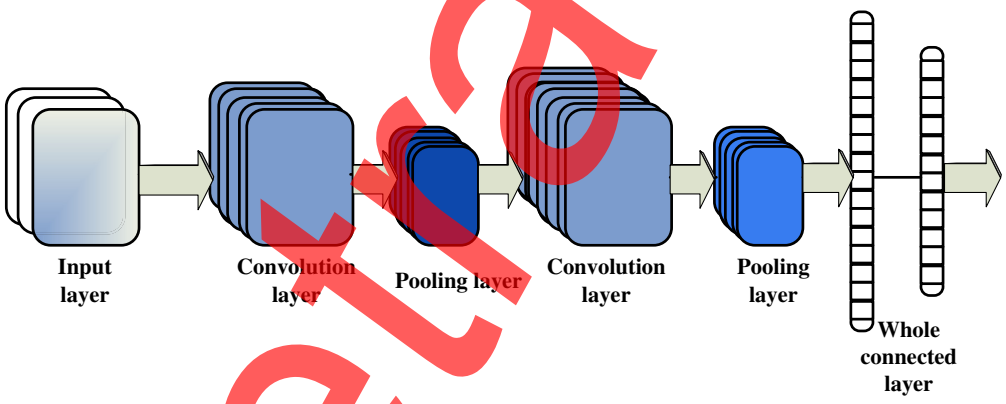


Fig. 1 CNN conceptual architecture.

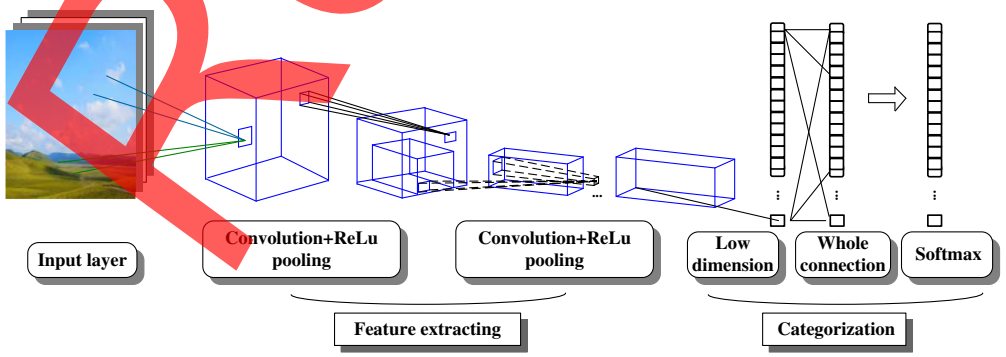
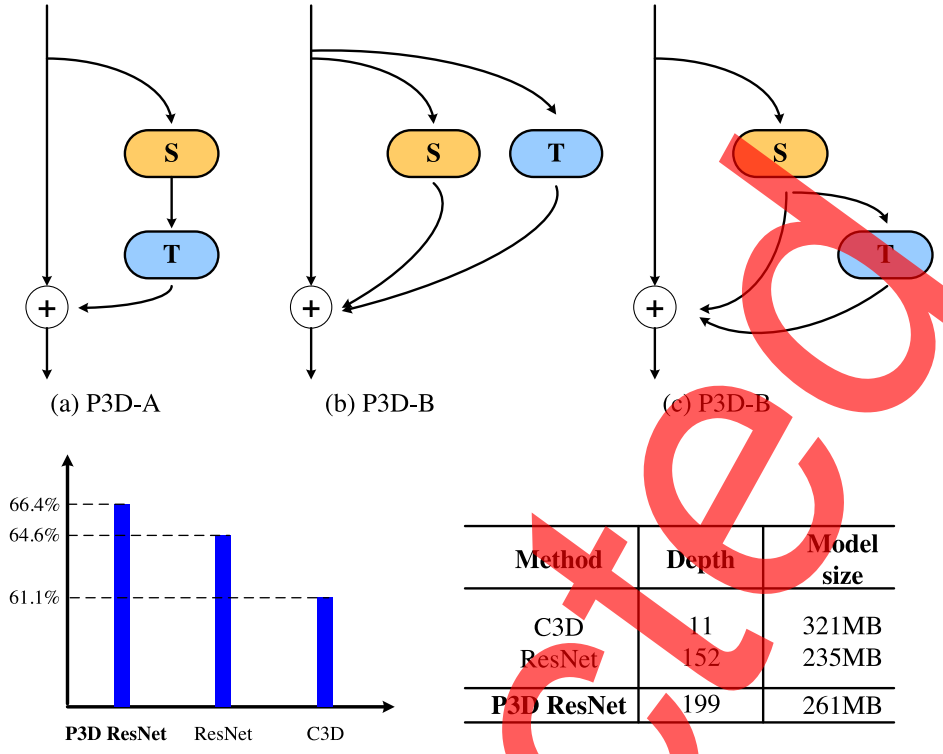


Fig. 2 CNN feature extractor.



Method	Depth	Model size
C3D	11	321MB
ResNet	152	235MB
P3D ResNet	199	261MB

Fig. 3 (a)–(c) Improved structure of three networks.

3.3 *Global Temporal CNN Cross-spectral Feature Construction*

This assumes a convolutional neural network with L layers and image I. The image$I$ is fed into the convolutional neural network of the L layer and the feature maps obtained by convolution of different kernels can be obtained.[11] Therefore, the convolutional feature map can be represented by the following formula:

$$\bar{F}_i = \{F_y : i = 1 \ldots L; j = 1 \ldots C_i\}. \tag{9}$$

Among them, F_{ij} represents the j'th feature map on the ith convolutional layer, and C_i represents the number of convolution channels, i.e., the number of convolution kernels. With these definitions, the global max pooling on the convolutional layer (finding the maximum response value on the j'th convolutional feature map) can be expressed by the following formula:

$$\dot{V}_{F_{i,j}} = \max(f_i(l_1, l_2)); l_1, l_2 \in F_{i,j}. \tag{10}$$

Then the global max pooling on each channel can be expressed as

$$\dot{V} = [\dot{V}_{F_{i,j}} : j = 1 \ldots C_i]. \tag{11}$$

Similarly, the global average pooling on the convolutional layer can be expressed in the form of the formula

$$\bar{V}_{F_{i,j}} = \frac{1}{W_i * H_i} \sum f_i(l_1, l_2). \tag{12}$$

Then the global average pooling on each channel can be expressed as

$$\bar{V}_i = [\bar{V}_{F_{i,j}} : j = 1 \ldots C_i]. \tag{13}$$

The max pooling operation characterizes the local maximum response of the feature map. Average pooling describes the overall response on the feature map. The research objects in the images generally appear in different shapes, spatial coordinates, and scales. The pure pooling operation does not fully consider the spatial and scale information of the target object in the image. Therefore, this paper proposes a multiscale pooling strategy to overcome the variation of the target in the scale space. The definition of the multiscale pooling strategy is as follows:

$$V_i = [V_{F_{i,j,R}} : j = 1 \ldots C_i], \quad V_{F_{i,j,R}} = P_{l_1, l_{2 \in R}} |f_i(l_1, l_2)|, \tag{14}$$

$$V_i = [V_{F_{i,j}} : j = 1 \ldots C_i], \quad V_{F_{i,j}} = \sum_{R \in F_{i,j}} V_{i,R}, \tag{15}$$

where R represents a square area on the feature map. The selection of four scale value parameters for the region $R: 1 \times 1$, 2×2, 4×4, and 7×7 show the construction process of the global time domain feature CNN feature. In the first step, the output feature map (the output of the last pooling layer) extracted from the video frame by the VGG-16 network is subjected to pooling operations of four different scales. It gets four feature maps: $(7 \times 7 \times 512)$, $(4 \times 4 \times 512)$, $(2 \times 2 \times 512)$, and $(1 \times 1 \times 512)$. Where 512 represents the number of channels

$$V^r = \sum_{g=1}^{4} V_8^r; r = 1, 2, 3, \ldots 512. \tag{16}$$

In action recognition, it is very important and meaningful to construct a suitable visual dictionary. Using the constructed visual dictionary, the global temporal features extracted from each video can be mapped to a unique visual word through Euclidean distance and nearest neighbor matching algorithm.[12] The probability distribution of each visual word in a video with N visual words w^* can be expressed by the following formula:

$$P(\theta, z, w^* | \alpha, \phi) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n^*|z_n, \phi). \tag{17}$$

In the above formula, θ represents the video-topic distribution, and each item in θ represents the probability of each topic appearing in the current video. α is the Dirichlet prior parameter used to generate the video-topic distribution, and Z_n represents the topic, obtained from θ. φ represents the topic-visual word distribution, while φ determines $p(w_i^*|z_n)$. That is, the probability that the visual word belongs to a certain topic (basic action).

The joint probability distribution of all visual words in a video can be obtained by summing the integrals θ and topics z

$$p(w^*|\alpha, \phi) = \int p\left(\theta|\alpha\left(\prod_{n=1}^{N} \sum_{Z_n} p(Z_n|\theta) p(w_n^*|Z_n, \phi)\right)\right) \mathrm{d}\theta. \tag{18}$$

The joint probability distribution of the entire video set can be obtained by multiplying all the videos by this formula. Where "D" represents the video set, and "M" represents the number of videos in the video set

$$p(D|\alpha, \phi) = \prod_{d=1}^{M} \int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d} \sum_{Z_{dn}} p(Z_{dn}|\theta_d) p(w_{dn}^*|z_{dn,\phi})\right) \mathrm{d}\theta_d. \tag{19}$$

Through the above steps, it completes the construction of a global feature CNN cross-spectral topic model for the video set. After that, it is necessary to use the Gibbs sampling algorithm to train the training video set to obtain the specific form of the CNN topic model and then determine the parameters in the model. There are four main steps in the training process: In the first step, it randomly assigns a topic to each visual word in each video in the video set. In the second step, it rescans the entire video set. It uses the Gibbs sampling formula for each word. It calculates the

probability that the currently sampled visual word corresponds to each topic. The third step is to update the topic of the currently sampled visual word according to the obtained probability distribution. In the fourth step, the second and third steps are repeated until the distribution of topic visual words and the distribution of video topics converge or reach a preset number of iterations. The formula for each sampling of the Gibbs sampling algorithm is

$$P(Z_i|Z_{-i},w)\frac{n_{m,-j}^{(k)}+\alpha_k}{\sum_{k=1}^{k}(n_{m,-i}^{v}+\alpha_k)}\cdot\frac{n_{k,-i}^{(v)}+\beta_v}{\sum(n_{k,-i}^{(v)}+\beta_v)}. \tag{20}$$

It counts the number of sight words in each topic and the number of sight words contained in each topic in each video. This estimates the probability distribution that the currently sampled visual word belongs to each topic and the probability distribution that the topic belongs to each document. The topic-visual word distribution and video-topic distribution θ can be easily obtained by this formula

$$\theta=\frac{n_{m,-i}^{(k)}+\alpha_k}{\sum_{k=1}^{k}(n_{m,-i}^{(v)}+\alpha_k)},\phi=\frac{n_{k,-i}^{(v)}+\beta_v}{\sum_{v=1}^{v}(n_{k,-i}^{(v)}+\beta_v)}. \tag{21}$$

After extracting the outgoing information, the next step is to use the above method to classify: it first clusters the descriptors to form visual words and thesaurus. It forms a lexical description of the video, and finally completes the action recognition through a classifier.[13]

After extracting the global temporal feature CNN feature from each frame of the video in the human action recognition database, the constructed visual dictionary generates a word frequency matrix corresponding to each video.[14] Through such conversion, a video can be regarded as a document. The word frequency matrix corresponding to the video is the number of times.

4 Cross-spectral Human Behavior Recognition Experiment Based on Deep Convolutional Network for Global Temporal Representation

To better verify the effectiveness of the algorithm proposed in this paper, two datasets composed of video clips intercepted from real life are selected: unified compliance framework (UCF)-sports dataset (as shown in Fig. 4) and UCF-11 (as shown in Fig. 5).

The videos in the UCF-11 database are the videos uploaded by netizens collected from the large video social networking site YouTube. The database has about 1600 videos containing 11 different action behaviors. Each type of action is divided into 25 parts according to different video shooting conditions, and each part has about four videos. This dataset is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, complex background, and lighting conditions. Part of the content in the database is shown in Fig. 5.

Fig. 4 UCF-sports database.

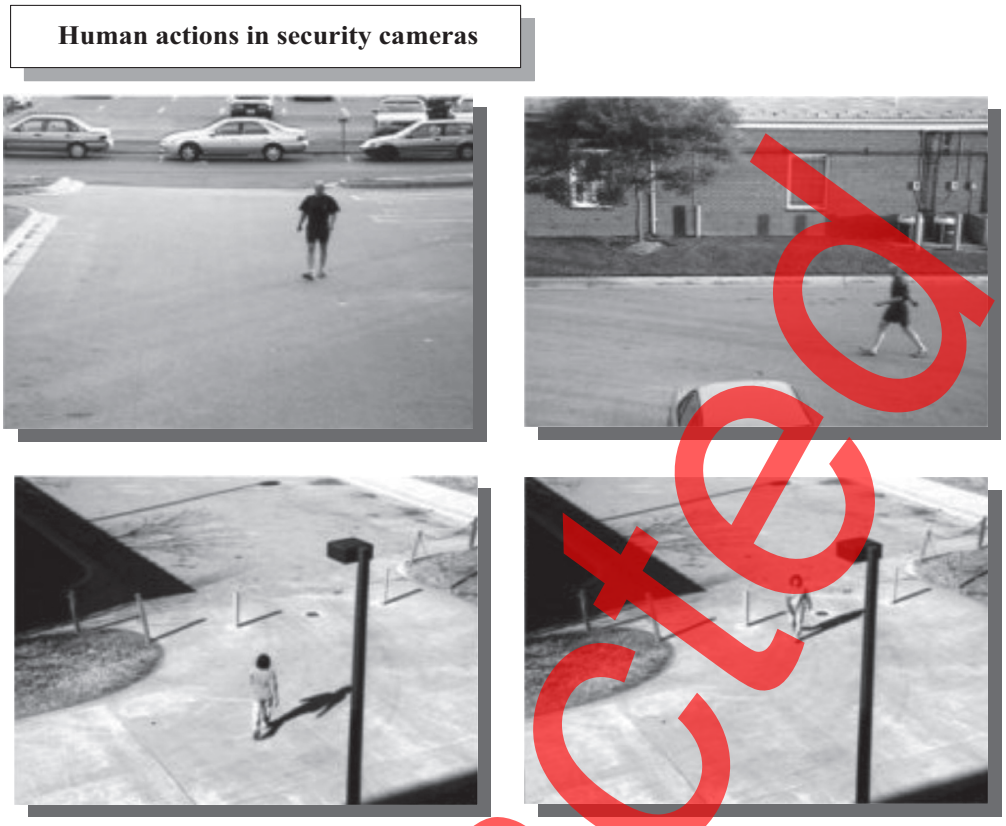


Fig. 5 UCF-11 database.

4.1 *Experimental Setup*

The hardware environment of this experiment is Intel® Core™ i5-6600k CPU @ 3.50 GHz 3.30, the memory size is 8 GB, the graphics card is NVidia GTX960, and the video memory size is 4G. The operating system is Windows 10 64-bit. The compilation environment is Python 2.7.[15] Several open-source software and machine learning toolkits are used: TensorFlow version 1.2.0; OpenCV version 2.4.0; Sklearn version 0.19. TensorFlow is a deep learning framework developed by Google researchers.[16] Many pretrained deep neural network models are provided in TensorFlow: AlexNet, VGG-Net, Google-Net, etc. The pretrained VGG-16 convolutional neural network used in this paper is implemented through the TensorFlow framework.[17] OpenCV is an open-source computer vision algorithm library mainly written in C language. OpenCV provides interfaces for a variety of development languages including Python, Ruby, MATLAB, etc. It supports the current mainstream operating systems MAC OS, Windows, Linux, etc.[18] OpenCV implements many classic algorithms for computer video and image processing. In this article, the video image reading and other related processing are all implemented through OpenCV. Sklearn is an elegantly designed machine learning algorithm library, which is a simple and effective tool for data mining and analysis.[19] Sklearn implements many classic machine learning algorithms. The SVM algorithm used in this article is implemented by calling the algorithm interface in Sklearn. In the experiment, the most important parameter setting is the determination of the number of topics in the laser doppler anemometer topic model. The specific experimental data is shown in Fig. 6.

When conducting experiments, the dataset needs to be divided into training set and test set. There are mainly the following methods for dividing the training set and the test set: one-to-one cross-validation, *k*-fold cross-validation, and random data splitting. In this experiment, accuracy is not the single parameter, however, the author has conducted other experiments in other parts of the passage and offered more parameters to be referred to.

The *k*-fold cross-validation method divides the original dataset into *k* groups. It uses a set of data as the test set and the rest as the training set to train the model. It repeats the training

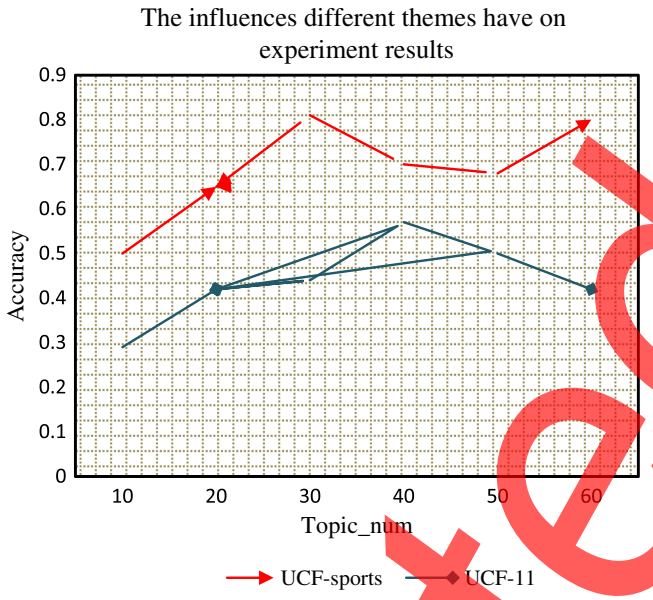


Fig. 6 Influence of the number of topics on the experimental results.

and testing of the model k times to ensure that each set of data is used as a test set. Finally, it calculates an average of the results of these k tests as the final experimental result of the model.[20] This assumes that the original dataset has Q data samples. De-one cross-validation is to use a sample in the original dataset as the test set. It takes the remaining Q-1 data samples as the training set. It repeatedly uses each sample as a test set to train and test the model Q times. It finally calculates the results of the Q tests to obtain a recognition rate as the recognition rate of the model. The method of randomly splitting data refers to repeating multiple experiments repeatedly when conducting experimental tests. In each experiment, the original dataset is randomly divided into training set and test set according to a certain proportion. It averages the results of multiple experiments as the experimental results of the final model.[21] Generally speaking, the one-to-one cross-validation method requires repeated cross-validation Q times, which is computationally expensive. Therefore, the deone cross-validation method is generally only suitable for cases with relatively small sample size. By comparing the above several dataset partitioning and verification methods, it is found that the k-fold cross-validation method is more reliable than the other two dataset partitioning methods for the algorithm.[22] In the experiments, k takes values of 6 (UCF-sports) and 10 (UCF-11). The random partition strategy is implemented internally by the TensorFlow open-source framework. The division ratio is 80% of the samples for training and 20% of the samples for testing. To make sure the results are valid, the author trained the random database for four times, and the results of the algorithm from the previous chapter appear physical.

To utilize CNN features to construct more effective and robust feature representations, various strategies are experimented for feature construction. Table 1 shows the experimental results

Table 1 Experimental results of different CNN feature construction strategies.

Algorithm	UCF-sports	UCF-11	Feature dimension
Fc7	93.0%	66.93%	6124
Max-pooling	94.4%	85.2%	1024
Average-pooling	95.0%	85.1%	1024
Multiscale-max-pooling	99.7%	83.6%	1024
Multiscale-average-pooling	86.7%	97.8%	1024

of several feature construction methods on the datasets UCF-Sports and UCF-11. "fc7" indicates that the feature of the video frame is the output of the second 4096-dimensional fully connected layer of the VGG-16 network. "max-pooling" means that the output of the fifth-pooling layer of the VGG-16 network is subjected to a single-scale maximum pooling operation to generate image features of a single frame. "Average-pooling" means that the output of the fifth-pooling layer of the VGG-16 network is subjected to a single-scale average pooling operation to generate image features of a single frame. "Multiscale-max-pooling" means that the output of the fifth-pooling layer of the VGG-16 network is subjected to a multiscale maximum pooling operation to generate image features of a single frame. "Multiscale-average-pooling" means that the output of the fifth-pooling layer of the VGG-16 network is subjected to a multiscale average pooling operation to generate image features of a single frame.

Through experimental comparison, it is found that the average pooling strategy based on CNN features has the best experimental performance among several feature construction strategies, and its feature dimension is also relatively low.

4.2 *Experimental Results of UCF-Sports Dataset*

Table 2 shows the experimental results of the algorithm proposed in this paper and the experimental results of most of the industry's classic traditional methods on UCF-sports. We can see that compared with the classical traditional methods, the algorithm proposed in this paper has a significant improvement in the accuracy of the algorithm. It is a 5% improvement compared to the highest algorithm.

Confusion matrix is an important means to analyze and evaluate the experimental results of the algorithm. In this paper, the confusion matrix reflects the recognition accuracy of each type of human behavior and the situation that human behavior is misclassified into other behavior categories. The confusion matrix can not only reflect which actions and behaviors are easy to identify errors, but also help analyze the reasons for the misidentification through the confusion matrix.

Confusion matrix is not the best way to compare the experimental data, but in this paper, it suits the algorithm best. Table 3 shows the confusion matrix of the experimental results of the algorithm proposed in this paper. The first column in the confusion matrix represents the true behavior category of the test dataset. The first row represents the behavior categories predicted by the proposed algorithm on the test dataset. The values of the diagonal elements in the table represent the proportion of correctly identified samples. The larger the value, the higher the accuracy of such human behavior recognition. The following analysis will start with the confusion matrix. [Capital letters in the table represent the different behaviors identified: A (diving), B (golfing), C (kicking), D (lifting), E (riding), F (running), G (skateboard), H (gymnastics 1), I (gymnastics 2), and J (walking)].

As can be seen from Table 3, the algorithm proposed in this paper has a recognition accuracy of eight types of human behaviors on the UCF-sports dataset that exceeds 90%. Among them, the recognition accuracy of diving, kicking, weightlifting, and gymnastics has reached 100%. It can also be found through the confusion matrix that the recognition accuracy of horse riding in the experiment is very low. It is only 58%, which is far lower than the recognition accuracy of

Table 2 Comparison of experimental results on UCF-sports dataset.

Algorithm	Accuracy
GL	76.2%
ISA	78.7%
HOG3D + HOF + HOG	79.6%
DT	90.2%
Methods of this article	95.0%

Table 3 Confusion matrix of experimental results on UCF-sports dataset.

	A	B	C	D	E	F	G	H	I	J
A	0.6	0.02	0.15	0.06	0.15	0.04	0.24	0.24	0.11	0.5
B	0.32	1.95	0.24	0.45	0.36	0.14	0.36	0.17	0.16	0.18
C	0.02	0.06	1	0.12	0.24	0.24	0.21	0.16	0.29	0.26
D	0.33	0.12	0.13	1	0.16	0.36	0.06	0.22	0.14	0.24
E	0.06	0.06	0.41	0.03	0.58	0.14	0.19	0.34	0.17	0.14
F	0.5	0.13	0.26	0.02	0.34	0.85	0.33	0.21	0.03	0.47
G	0.24	0.15	0.18	0.45	0.24	0.36	0.92	0.32	0.05	0.11
H	0.46	0.18	0.14	0.15	0.29	0.15	0.25	1	0.5	0.24
I	0.08	0.04	0.03	0.36	0.24	0.24	0.29	0.24	0.83	0.36
J	0.17	0.03	0.14	0.25	0.39	0.34	0.24	0.14	0.18	0.88

other behavioral categories. The main reasons for this analysis are as follows: first of all, compared with the pure human behavior, the action behavior of horse riding is more difficult to recognize due to the addition of animals. And the number of video samples for the horse riding category is only 12. It occupies a small proportion in the entire UCF-sports database, which largely affects the recognition accuracy of the experiment. Second, the relative time length of the video of horseback riding is short, which also affects the recognition accuracy of the experiment to a certain extent. Through the confusion matrix, it can also be seen that the behavior recognition accuracy of walking is also very low. It is only 77%, which is the second-lowest recognition accuracy after horseback riding. After research to identify the wrong video data, it is found that there are two main reasons for the low recognition rate: the first is that the lighting environment of the video scene is relatively poor, which leads to less contrast between people and the surrounding environment. It is difficult to extract effective video motion features. This is also the main reason for the low recognition rate. Second, due to the shooting angle, some people have a lower proportion in the video. When the model is extracting features, the response area is smaller. This is also an important reason for the low accuracy of walk recognition.

The proposed human body recognition algorithm based on global temporal features CNN features achieved an average accuracy of 90.0% on the UCF-sports database. UCF-sports data collects a large number of human behavior videos in various life and sports scenarios. Behavior categories range from simple walking and running to complex horseback riding and gymnastics. So it is very challenging to perform experiments on this database. The experimental results obtained on this dataset also verify the effectiveness of this algorithm in human action recognition tasks.

4.3 *Experimental Results on UCF-11 Dataset*

Compared with the UCF-sports dataset, the UCF-11 dataset is more complex in human behavior categories. Most of them are complex behaviors such as shooting, volleyball spiking, and tennis. The overall recognition effect on the UCF-11 dataset is not high. As shown in Table 4, the algorithm proposed in this paper achieves an average accuracy of 70.1% on this dataset. Compared with most traditional algorithms, the algorithm proposed in this paper has further improved the recognition accuracy.

As with the analysis of the experimental results on the UCF-sports dataset, the experimental confusion matrix is also analyzed on the UCF-11 dataset. The content of the confusion matrix is shown in Table 5. The confusion matrix shows the recognition rate of 11 different human behaviors. Through the confusion matrix, it can be found that among the 11 human behaviors, the recognition rate of three behaviors exceeds 90%. There are two behaviors with a recognition

Table 4 Comparison of experimental results on UCF-11 dataset.

Algorithm	Accuracy
IL-unlabeled	48.6%
HAR + HES + MSER	69.5%
perOF + perHOG + objOF + objHOG + GIST + COLOR	69.8%
Methods of this article	85.1%

Table 5 Confusion matrix of experimental results on UCF-11 dataset.

	A	B	C	D	E	F	G	H	I	J	k
A	0.79	0.11	0.18	0.47	0.24	0.14	0.16	0.37	0.36	0.15	0.26
B	0.02	0.81	0.18	0.36	0.36	0.23	0.25	0.24	0.25	0.09	0.34
C	0.34	0.65	0.79	0.34	0.46	0.56	0.34	0.06	0.14	0.15	0.15
D	0.26	0.14	0.56	0.86	0.23	0.36	0.39	0.09	0.26	0.36	0.09
E	0.14	0.33	0.14	0.45	0.8	0.36	0.26	0.19	0.25	0.39	0.28
F	0.01	0.07	0.11	0.26	0.07	0.71	0.33	0.25	0.48	0.18	0.33
G	0.06	0.26	0.09	0.27	0.08	0.15	0.83	0.36	0.28	0.06	0.17
H	0.17	0.31	0.36	0.38	0.36	0.37	0.42	0.81	0.34	0.28	0.12
I	0.45	0.14	0.32	0.26	0.16	0.25	0.38	0.08	0.64	0.39	0.1
J	0.41	0.23	0.15	0.05	0.23	0.19	0.39	0.69	0.16	0.78	0.26
K	0.15	0.18	0.48	0.09	0.05	0.23	0.31	0.23	0.24	0.18	0.61

accuracy of >80%, with the highest recognition accuracy of 93%. Three behaviors were identified with <75% accuracy: jumping, walking with the dog, and playing soccer. By identifying some of the wrong behaviors, the reasons for the low recognition results of these three behaviors will be analyzed in detail below. [Capital letters in the table represent the different behaviors identified: A (shooting), B (biking), C (diving), D (golf), E (horse riding), F (playing soccer), G (swing), H (tennis), I (jumping), J (volleyball spike), and K (walking with the dog)].

Among the three behaviors with low recognition accuracy, the recognition rate of jumping action is the lowest, only 64%. Only about half of the jumping motions were recognized correctly in the 100 test videos. A study of the misidentified videos in the test videos found that jumping motion varied considerably from one video to the next. The recognition accuracy of walking motion is only 61%, which is also much lower than other motion recognition results. The walking action is mainly the action behavior of a person walking a dog while walking, and it is a very complex action behavior. Since the jumping action is an action on a trampoline, there are upright jumping, lying flat jumping, and other different forms of jumping. It also includes the jumping action of many people, and the body also partially blocks the action. Also, the scene in most videos is dimly lit. These factors together affect the accuracy of recognition.

The misrecognized videos were distributed among the actions of shooting, cycling, diving, and golf. By observing the training video and the test video, it is found that the algorithm's subject histogram for the behavior of walking with a dog is quite different. This shows that the model algorithm has not learned a feature expression with strong discrimination for this action. This is also where improvements will be made in the future.

The action behavior in the UCF-11 database is more complex, and the scene information is richer. Experimenting on this database is an important test of the practicability and effectiveness of the algorithm model. The algorithm proposed in this paper also achieves an average accuracy

of 76.6% on this database, which is higher than the accuracy of the classical methods. This further verifies the effectiveness of our algorithm in recognizing human behavior.

4.4 *Experiment of Cross-spectral Human Behavior Recognition Algorithm Based on Global Time-domain Feature CNN*

The main content of this section is the relevant experimental results of the human action recognition algorithm based on the global time domain feature CNN. The method of this paper can have a great influence upon the experiment; however, it needs to be adjusted according to the existing configuration. Considering the number of experimental samples and the problem of experimental hardware settings, the UCF-sports dataset has only 150 videos, which is not conducive to the training of deep models. Therefore, the effectiveness of our proposed human action recognition algorithm based on the global temporal feature CNN is only verified on the UCF-11 dataset (1600 videos). During model training, we first normalize the images of the input video frames to a size of 224×224 pixels. It then sets batch_size to 64 with an initial learning rate of 0.001. It is set that every 5000 steps of training, the learning rate becomes one-tenth of the original.[23] It initializes the CNN module with VGG-16 model parameters pretrained on ImageNet. Figure 7 shows the convergence of the loss function of the model initialized with and without pretrained weight parameters.

From Fig. 7, we can find that using the pretrained weight parameters to initialize the model not only converges faster but begins to decline at about 20 cycles, and the loss function can converge to a stable minimum value. When the model is not initialized with pretrained weight parameters, the loss function not only falls slowly. And the loss function has been oscillating around 0.2 in the end. So we use the VGG-16 network model parameters pretrained on ImageNet as parameters to initialize the CNN part of the model.

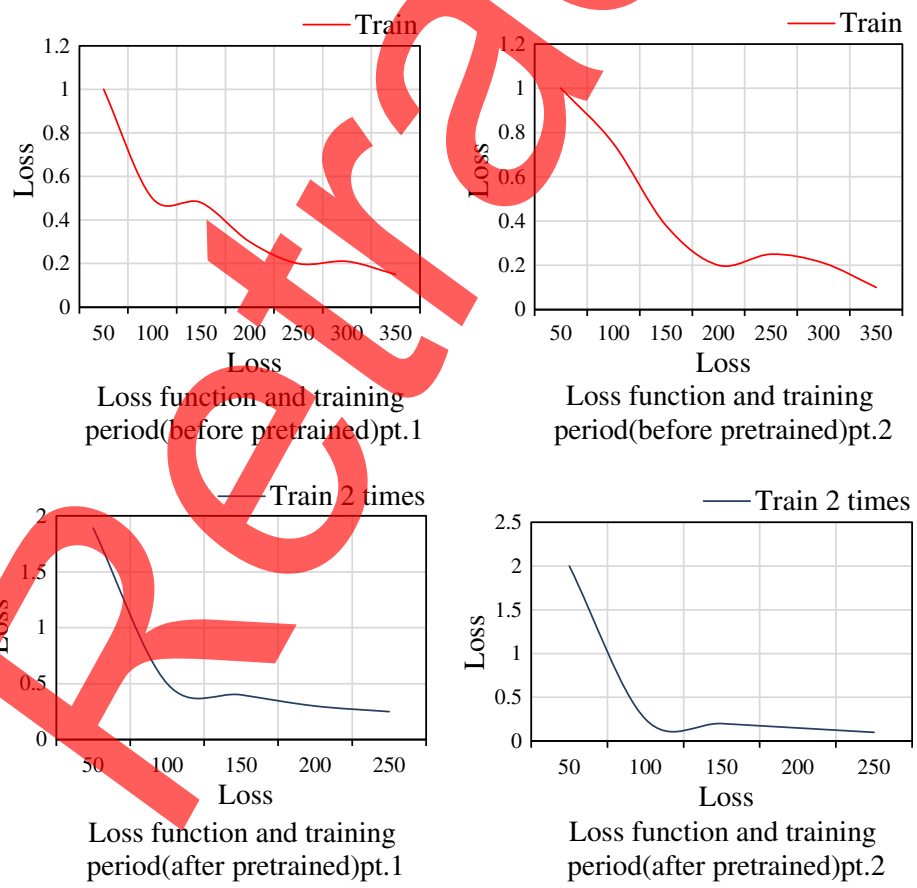


Fig. 7 The relationship between the loss function and the training period when initializing network training with a pretrained model.

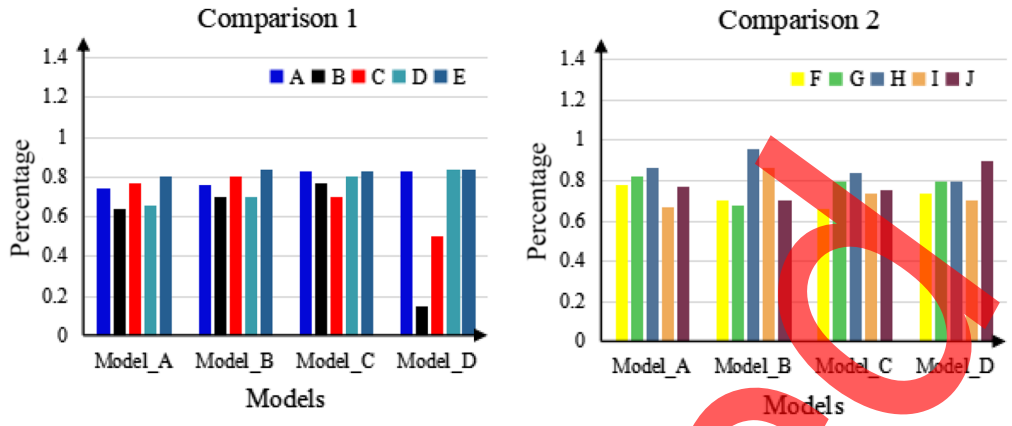


Fig. 8 Experimental results of different models on UCF-11.

Figure 8 shows the experimental results of different models on UCF-11. Where capital letters represent the different behaviors identified [A (shooting), B (biking), C (diving), D (golf), E (horse riding), F (football), G (swing), H (tennis), I (jump), and J (volleyball spike)]. Model_D represents the result of the cross-spectral human action recognition algorithm based on the global time-domain feature CNN proposed in this paper. Model_B indicates that the 512-dimensional feature vector obtained after the output of the last pooling layer of VGG-16 is subjected to the global average pooling operation is input into LSTM, i.e., single-scale CNN feature recognition. Model_C represents the experimental results of the human action recognition algorithm with global temporal feature CNN without adaptive fusion strategy. It uses the same feature fusion method as Model_A. Model_A represents the experimental results of the human action recognition algorithm using the global temporal feature CNN of the adaptive fusion strategy. From the figure, we can see that Model_D achieved the highest accuracy of 90%. Model_D improves the accuracy by about 16% compared with Model_A. Model_D improves the accuracy by about 6% over Model_C. Model_C improves the accuracy by about 5% over Model_B. This fully verifies the effectiveness of our proposed global temporal feature CNN for cross-spectral human action recognition algorithm.

Figure 9 shows the relationship between the fusion weight parameter and the running period. w1 represents the weight parameter of the 7×7 scale feature, and w2 represents the weight parameter of the 4×4 scale feature. $w3$ represents the weight parameter of the 2×2 scale feature, and w4 represents the weight parameter of the 1×1 scale feature. Where $w1 = 0.331$, $w2 = 0.214$, $w3 = 0.215$, and $w4 = 0.217$. We can see that the 7×7 scale features contribute the most to the entire parameter fusion, and the weights of other scale features are not much different. This further illustrates that the overall characteristics of people play a great role in recognizing human behavior. This is similar to human thinking. When recognizing human actions, it first looks at the whole and then uses some detailed information to assist the overall judgment.[24]

5 Conclusions

This paper introduces a cross-spectral human action recognition algorithm based on the global temporal feature CNN feature. It also experimented with several CNN feature-based human action recognition feature construction methods. It takes two datasets UCF-sports and UCF-11 content pictures containing many different types of actions as recognition objects. They have achieved higher average accuracy than other algorithms, respectively. Through experiments on the algorithm model, it fully verifies the effectiveness of the algorithm proposed in this paper in recognizing human behavior. However, there are still many deficiencies in this article. Due to the limited space, the author did not summarize the method in particular and the derivation process was not rigorous enough. In the future, the author's goal is to further optimize this algorithm to apply it to more behavior recognition, and also to conduct research on faster and more accurate algorithms for behavior recognition, so that they can be used in more practical applications in daily life scenarios.

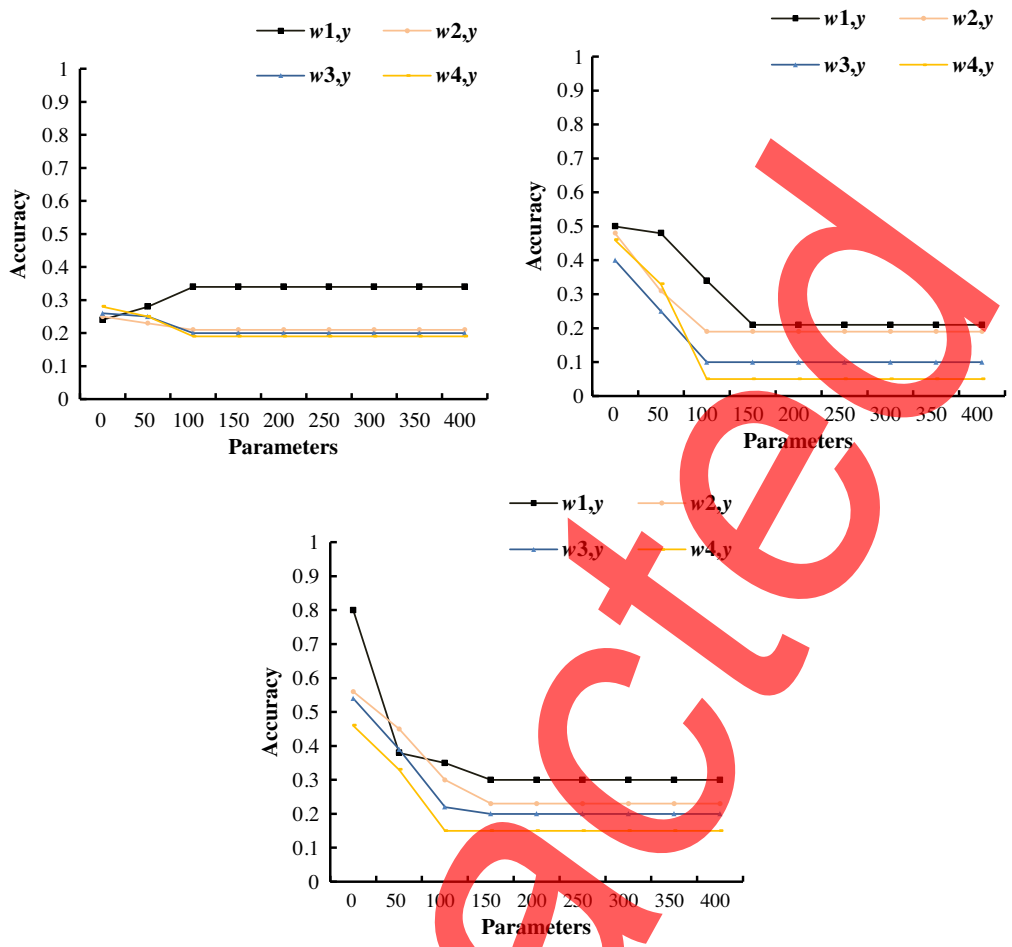


Fig. 9 The relationship between the fusion weight parameter and the running period.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Award Nos. 62066032 and 61866006); Natural Science Foundation of Guangxi Province (Award No. 2021GXNSFAA075019); Project of National Ethnic Affairs Commission of China (Award No. 2020-GMI-010); Guangxi innovation-driven development of special funds project (Award No. Gui Ke AA17204091); Vocational Education Teaching Reform Research Project of Guangxi (Award No. GXGZJG2019A045); Middle-aged and Young Teachers' Basic Ability of Scientific Research Promotion Project of Guangxi (Award No. 2021KY0130); Philosophy and Social Science Foundation of Guangxi (Award No. 21FYJ041); Higher Education Undergraduate Teaching Reform Project of Guangxi (Award Nos. 2021JGA243 and 2022JGB175); The Open Research Fund of Guangxi Key Lab of Human-machine Interaction and Intelligent Decision (Award No. GXHIID2213). This study acknowledge the support of the Logistics Engineering Innovation Laboratory, Logistics Engineering Technology Laboratory and Smart Logistics Exhibition Center of Nanning Normal University. The authors gratefully acknowledge the support of Construction project of Practice conditions and practice Base for industry–university co-operation of the Ministry of Education (Award No. 202102079139).

References

1. X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *Int. J. Comput. Vision* **123**(3), 309–333 (2017).

2. B. Li, W. Guo, and X. L. Zhang, "A global manifold margin learning method for data feature extraction and classification," *Eng. Appl. Artif. Intell.* **75**(Oct.), 94–101 (2018).
3. W. Xie, Y. Li, and X. Jia, "Deep convolutional networks with residual learning for accurate spectral-spatial denoising," *Neurocomputing* **312**(Oct. 27), 372–381 (2018).
4. Z. Zhao, S. Srkk, and A. B. Rad, "Kalman-based spectro-temporal ECG analysis using deep convolutional networks for atrial fibrillation detection," *J. Signal Process. Syst.* **92**(7), 621–636 (2020).
5. P. Xiao, Y. Guo, and P. Zhuang, "Removing stripe noise from infrared cloud images via deep convolutional networks," *IEEE Photonics J.* **10**(4), 1–14 (2018).
6. L. Wen, F. Jia, and Q. Hu, "Automatic segmentation of liver tumor in CT images with deep convolutional neural networks," *J. Comput. Commun.* **03**(11), 146–151 (2017).
7. X. Yang, Z. Zeng, and S. Yi, "Deep convolutional neural networks for automatic segmentation of left ventricle cavity from cardiac magnetic resonance images," *IET Comput. Vision* **11**(8), 643–649 (2017).
8. T. Nguyen, V. Bui, and G. Nehmetallah, "Computational optical tomography using 3-D deep convolutional neural networks," *Opt. Eng.* **57**(4), 043111 (2018).
9. X. Yuan and P. U. Yunchen, "Parallel lensless compressive imaging via deep convolutional neural networks," *Opt. Express* **26**(2), 1962–1977 (2018).
10. S. Hong, S. Kwak, and B. Han, "Weakly supervised learning with deep convolutional neural networks for semantic segmentation: understanding semantic layout of images with minimum human supervision," *IEEE Signal Process. Mag.* **34**(6), 39–49 (2017).
11. R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Vehicle lane markings segmentation and keypoint determination using deep convolutional neural networks," *Multimedia Tools Appl.* **80**(12), 1–15 (2021).
12. F. I. Eyiokur, D. Yaman, and H. K. Ekenel, "Domain adaptation for ear recognition using deep convolutional neural networks," *IET Biometrics* 7(3), 199–206 (2018).
13. Z. Kolar, H. Chen, and X. Luo, "Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images," *Autom. Construct.* **89**(May), 58–70 (2018).
14. Y. Sun et al., "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.* **24**(2), 394–407 (2020).
15. Y. Wu and J. Liu, "Research on college gymnastics teaching model based on multimedia image and image texture feature analysis," *Discov. Internet Things* **1**, 15 (2021).
16. B. Chikhaoui, B. Ye, and A. Mihailidis, "Feature-level combination of skeleton joints and body parts for accurate aggressive and agitated behavior recognition," *J. Ambient Intell. Hum. Comput.* **8**(6), 957–976 (2017).
17. S. M. Aljaberi and A. N. Al-Masri, "Automated deep learning based video summarization approach for forest fire detection," *J. Intell. Syst. Internet Things* **5**(2), 54–61 (2021).
18. A. Lentzas and D. Vrakas, "Non-intrusive human activity recognition and abnormal behavior detection on elderly people: a review," *Artif. Intell. Rev.* **53**(3), 1975–2021 (2020).
19. M. Quaid and A. Jalal, "Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm," *Multimedia Tools Appl.* **79**(9–10), 6061–6083 (2020).
20. P. Schrter and S. Schroeder, "The Developmental Lexicon Project: a behavioral database to investigate visual word recognition across the lifespan," *Behav. Res. Methods* **49**(6), 2183–2203 (2017).
21. D. K. Jain et al., "GAN-Poser: an improvised bidirectional GAN model for human motion prediction," *Neural Comput. Appl.* **32**(18), 14579–14591 (2020).
22. M. Z. Uddin and J. Kim, "A robust approach for human activity recognition using 3-D body joint motion features with deep belief network," *KSII Trans. Internet Inf. Syst.* **11**(2), 1118–1133 (2017).
23. S. Lathuiliere et al., "Neural network reinforcement learning for audio-visual gaze control in human-robot interaction," *Pattern Recognit. Lett.* **118**(Feb.), 61–71 (2018).
24. N. Baker et al., "Local features and global shape information in object classification by deep convolutional neural networks," *Vision Res.* **172**(3), 46–61 (2020).

Xiaomo Yu is a doctor of mechanical automation, associate professor at Guangxi Key Lab of Human-Machine Interaction and Intelligent Decision, Nanning Normal University. His research interests include the internet of things, AI, machine learning, computer vision, emerging automation, logistics engineering, and management science and engineering. He has some application in the fields of metal machining, logistics systems optimization, brain computer interface technology, machine perception, and vision.

Xiaomeng Zhou is a postgraduate at Management Science and Engineering in Nanning Normal University. His research interests include the internet of things, logistics planning, and cloud computing.

Wenjing Li is currently a computer science professor at Guangxi Key Lab of Human-machine Interaction and Intelligent Decision, Nanning Normal University. His research interests include computer architecture, multicore processors, distributed parallel computing, and image processing.

Xinquan Liu is a doctor of traffic engineering and professor. He graduated from Southwest Jiaotong University in 2010. He worked at Nanning Normal University. His research interests include transportation management, traffic systems optimization, logistics planning, transport planning, logistics systems optimization, and travel behavior.

Peihua Song is a doctor of software engineering and lecturer. He graduated from Tongji University in 2019. He is worked at Nanning Normal University. His research interests include computer graphics and computer optimization techniques.