Basketball posture recognition based on HOG feature extraction and convolutional neural network

Jian Gao[1,*]

[1]Physical College, Zhengzhou University of Industrial Technology.
No.16, Xueyuan Road, Zhengzhou City, Henan Province, 451150, China

Abstract

Basketball posture recognition is one of the important research topics in human-computer interaction and physical education, which is of great significance in medical treatment, sports, security and other aspects. With the development of machine learning, the application value of basketball pose recognition in physical education is becoming more and more extensive. This paper constructs a novel convolutional neural network model to recognize basketball posture. The model consists of 11 layers. Convolution and pooling operations are carried out for five basketball postures in the sampled data set. By fusing with the features extracted from HOG, finer features can be obtained. Finally, the data set is trained and recognized by entering the full connection layer for classification. The results show that compared with the traditional machine learning methods, the recognition performance of new model is better.

*Corresponding author. Email: 7006 5024@qq.com

1. Introduction

At present, with the rapid development of human-computer interaction technology, human posture recognition technology is more and more attention. Posture recognition, as an important part of human behavior recognition, has become an important research focus in computer vision field in recent years. The main research method is to analyze the input parameters of whole or part of human limbs [1], such as human body contour, junction position, gesture and limb.

At the same time, human posture recognition has a wide range of application prospects, mainly used in the following aspects.

1) Intelligent human-computer interaction.

Through the recognition of human expression, posture or gesture to understand the human intention, so that the machine can recognize the human intention and make response to achieve the purpose of interaction [2,3].

2) Biometric recognition.

Through the analysis and recognition of people's behavior, posture, gait and other information, we can judge the specific attributes of people, which can be applied to identity identification [4,5].

3) Games and entertainment.

Users can interact with games through their own actions, which can bring new game experience to users. Users can also exercise while playing games, which is beneficial to people's health [6,7].

4) Auxiliary teaching.

Posture recognition of specific users can assist users in learning specific actions. For example, posture

recognition of athletes can determine whether a specific posture meets the standard [8,9].

The existing attitude recognition methods mainly include two kinds, one is human posture recognition based on image analysis, the other is human posture recognition based on motion sensor. Sensor-based recognition technology mainly allows researchers to carry sensors to collect relevant motion data, commonly used sensors mainly include accelerometer, reluctance sensor, gyroscope, etc.

After the motion information of researchers is acquired by sensors, the human posture is recognized by combining relevant methods of machine learning, such as Naive Bayes, improved random forest and support vector machine (SVM) [10-12]. The attitude recognition result of this method is mainly affected by the feature extraction method, i. e. sensor use and classifier selection. In addition, image-based analysis method is used to extract researchers' images as features of research and analysis. At present, image-based methods mostly use heap image aspect ratio, shape complexity change, eccentricity and so on to analyze the contour features of images combined with K-means or SVM to distinguish human pose categories.

Traditional machine learning methods mainly use linear discriminant functions to analyze and classify data, but it is often difficult to achieve good classification effect on a large number of complex and similar samples. However, deep learning network, with its strong autonomous learning ability and highly nonlinear mapping, can still achieve very good classification and recognition effect on some complex high-precision classification problems. It has been widely used in speech recognition, face recognition, image target classification and detection and other fields.

Although there are many research references on human pose recognition, the significance of this study lies in that, for static human pose images, the autonomous learning ability of deep convolutional network is used to extract image features instead of manually designed features [13]. Features extracted autonomously by network can provide more accurate feature representation than manual features. Convolutional network has complex nonlinear transformation ability and can mine data, namely image deeper information. At the same time, network has achieved excellent performance in the field of computer vision due to its characteristics of shared weight and invariable shift [14].

2. Convolutional neural network (CNN)

Convolutional neural network is developed from forward neural network and is mainly used in computer vision processing. Convolutional neural networks also belong to

forward neural networks. The difference lies in the different connection modes between layers of convolutional networks [15-18].

The common way forward neural networks connect is called "full connections," because the neurons in the hidden layer connect to all the neurons in the upper layer. Convolution network realizes the connection between network layers through convolution operation.

Convolutional neural network (CNN) has been proved to be a very efficient technique in various fields of pattern recognition. For example, GoogleNet has achieved very significant recognition effect in large-scale visual recognition. This network has 27 layers, mainly using maximum and average pooling, random inactivation, Softmax classifier, etc. At present, the network structure is still improving. In terms of deepening the network, the Residual module was proposed to prevent gradient dissipation. Feature learning is enhanced by jumping connections. In terms of widening networks, the Inception module [19] is proposed to extract multi-scale information of image through convolution operations of different branches. Besides, the Inception-V4 structure, which is formed by introducing jumping connections into Inception, can greatly accelerate the training speed of Inception model and improve the performance of network. In addition, features are processed and integrated at multiple scales, and a pyramid model composed of bottom-up and top-down repeated processing and intermediate supervision is adopted to improve the performance of the network, and excellent recognition results are obtained in attitude recognition.

The application of convolutional neural network in image recognition can be divided into two processes: training process and verification process. The training process is shown in figure 1. Image data sets are provided to the network as input features. After a series of convolution and pooling operations, the network will enter the full connection layer, whose purpose is to map "distributed feature representation" to the sample marker space. The mapped features can be classified by Softmax classifier. Classification results and data set labels calculate loss functions. And then the network parameters are adjusted by backward propagation through gradient descent and other network optimization algorithms. In the course of continuous training, loss is reduced to network convergence.

At this point, the network training process is complete. The verification process of the network is to cross-verify the trained network model. Part of the sample data is randomly selected as training data and provided to the network model for identification. The overall identification performance of the network is calculated by analyzing the success and failure of the model classification identification.
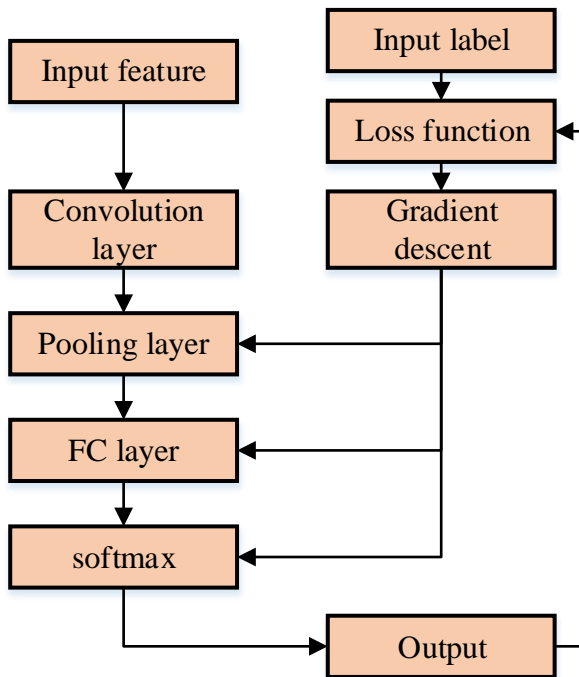
Figure 1. Training process of CNN

3. Proposed recognition structure

The proposed network structure has 11 layers, including 4 convolutional layers, 4 pooling layers and 3 fully connected layers. The feature maps of each convolution layer and pooling layer are different. The standard normal distribution is used to initialize the network. BP back propagation algorithm is used to train the network. Adam is used to optimize the network, and a total of 50 times are trained to minimize the cost function. In each iteration, 64 samples are randomly selected from the dataset. The sample images are uniformly converted into 100×100 pixels, and the network is trained iteratively at a learning rate of 0.0001. The following describes the details of network parameters and configuration.

3.1. Data enhancement and preprocessing

When the original image data set is not sufficient, data enhancement can be used to improve the data set, so as to improve the overall performance of the training network. The main methods of data enhancement are rotation, horizontal flip, mirror, shrink, put, random cutting, etc., and can be combined with a variety of processing methods. For example, rotate and scale at the same time. Rotation, mirroring, clipping, and their combination are used here. Data preprocessing mainly includes mean removal, normalization, PCA and so on.

To de-mean means to center all dimensions of the data to zero. Normalization is the normalization of data amplitude to the same range. PCA represents the number,

according to dimensionality reduction. Since the input is an image, the relative ranges of pixels are already approximately the same and are in the range 0 to 255, so this preprocessing step is not necessary. Here, only pixel size is normalized, without further de-mean and channel dimension normalization.

3.2. Convolution method

The convolutional operation distinguishes convolutional networks from the original fully connected neural networks. The convolutional operation provides an idea of weight sharing, that is, the features extracted from the convolutional network are the information in several local areas of the image rather than the information in a single pixel point. The convolution kernel w and bias b are initialized with a normal distribution of standard deviation 0.1. The convolution kernel of the first convolution layer has a size of 5×5, step size of 1, and a filling method of 0. The data obtained after passing through the first convolution layer with bias and entering the convolution layer have the same dimension, and then through the nonlinear activation function ReLU. ReLU function used in this paper is as follows:

$$f(x) = \max(0, w^T x + b) \quad (1)$$

Where x is the input of the first convolution layer. $w^T x + b$ represents the output after layer 1 feature mapping with bias. This article also tries to focus on different activation functions such as tanh and Leaky ReLU. The results show that this activation function can produce better results in training.

In the first convolution layer, a total of 32 feature maps are generated, and each feature map generates 100×100 output. In the second convolutional layer, namely, the third layer of the network, the 5×5 convolutional kernel is still used, with step size of 1 and filling mode of 0, and a total of 64 feature maps are generated. Each feature map produces an output of 50×50. In the third convolution layer, namely, the fifth layer of the network, a 3×3 convolution kernel with step size of 1 and filling mode of 0 is used to generate a total of 128 feature maps. Each feature map produces an output of 25×25. At the fourth convolutional layer, namely, the seventh layer of the network, a 3×3 convolutional kernel with step size of 1 and filling mode of 0 is used to generate 256 feature maps. Each feature map produces an output of 13×13. The data after each convolutional layer is transferred to the corresponding pooling layer for pooling feature mapping. The purpose of convolution operation is to improve the depth of feature extraction. Subsequent pooling operations aim to compress features and reduce parameters.

3.3. HOG feature extraction

Image preprocessing mainly includes the extraction of human posture ROI and normalization.

Extraction of human posture ROI

Firstly, the image is grayed, and the single-channel image can reduce the computation and facilitate feature extraction. The gamma coefficient is selected as 0.5 for gamma correction processing to weaken the interference of light intensity and color on HOG feature extraction in the later stage. Then, median filtering is performed on the gray-scale images, and Gaussian filtering is performed to eliminate salt and pepper noise and Gaussian noise, and reduce the influence of noise in the training samples on the results [20,21].

After denoising, the image is binarized. Threshold segmentation adopts maximum variance between categories as the method of threshold selection, which can well extract the human pose contour from the picture. Because the pixel values of people and the surrounding environment are quite different [22]. The pixel value of human posture area after contour extraction is 255, and the pixel value of background area is 0. Finally, morphological operation is performed on the image. This paper adopts the top hat operation mode. The top hat operation can fill the area with the largest pixel value among the 8 pixels around the pixel in the area with the low pixel value in the human posture area of the image, that is, the pixel value is set to 255, so as to make the human contour clear and complete.

Normalization of human motion areas

Firstly, the morphological processing results (target human body region) are marked, the area distribution of the marked region is counted, and the pixel value of the marked region is normalized.

HOG feature extraction

For normalized samples, all images in the four types of samples are processed in batches. First, the color space is normalized, and the gradient value and Angle value of each pixel in horizontal and vertical directions are calculated respectively. The total gradient value is the L2 norm of horizontal gradient and vertical gradient. Then, the histogram of gradient distribution is normalized, and the gray image with the adjusted size of 64*128 is divided into 8*8 pixel units. The gray-scale images of 180*128 are segmented into 9 equal parts on average. The gradient histogram of the nine directions is counted in units per cell. 242 cells form an alternate block, each block has 16*9 dimensional features. Each image is represented as 3780 dimensional features. Finally, histogram features of

gradient direction are obtained and sent to the subsequent neural network for training.

3.4. Pooling method

The essence of Pooling is sampling. Pooling is a method to compress the input feature map. The results of Pooling result in the reduction of features and parameters. However, the purpose of Pooling is not only this, but also to maintain certain invariance (such as rotation, translation, expansion, etc). The common pooling methods mainly include maximum pooling and mean pooling. Mean pooling means averaging the feature points in the field. Maximal pooling means maximal feature point. Feature extraction is mainly affected by two factors :1) the variance of estimated value increases due to the limitation of neighborhood size; 2) The parameter error of convolution layer causes the deviation of the estimated mean value. Generally speaking, mean-pooling can reduce the first error and retain more image background information, while max-pooling can reduce the second error and retain more texture information. Therefore, maximum pooling is adopted here, and the filling method is no filling, as shown in figure 2.
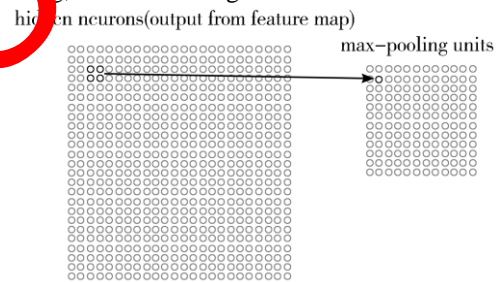

Figure 2. Pooling operation

It uses a 2×2 maximum pooling filter to implement down-sampling at step 2. In each pooling layer, the step is 2 for down-sampling, and the filling method is no filling. The result of down-sampling of the first convolution layer is 50 ×50. Subsequently, the results generated by the down-sampled convolution layer are 25×25, 13×13 and 6×6 respectively. After four pooling operations, the feature dimension is reduced to 6×6. Then, flatten it to the fully connected layer. In the whole convolutional neural network, the fully connected layer (FC) plays the role of "classifier".

3.5. Optimization method

The training of convolutional networks often requires a lot of time and resources, which is also an important

reason for bothering the development of network learning algorithms. Although the parallel distributed system can speed up the training of network model, it does not reduce the consumption of resources, and a good optimization algorithm can speed up the learning rate and reduce the consumption of resources. At present, the commonly used optimization algorithms are mainly the following.

1) SGD gradient descent method.

It is the most common optimization algorithm and a general optimization algorithm in the early stage. It uses the same learning rate for all parameter updates. For sparse data or features, people often want to update frequently occurring features more slowly. In this case, this optimization method is difficult to meet, in addition, this optimization algorithm is easy to converge to the local optimal solution.

2) Momentum.

It simulates the concept of momentum in physics, adding up the previous momentum instead of the actual gradient. Momentum term can accelerate SGD in the relevant direction, suppress oscillation, and thus accelerate convergence.

3) Nesterov.

Nesterov makes a correction during gradient updates to avoid moving too fast while improving sensitivity.

4) Adagrad.

Adagrad is actually a constraint on the learning rate, which is suitable for handling sparse gradients. However, its adjustment to the gradient is limited by a manually set parameter, namely the global learning rate, whose size will affect the adjustment rate.

5) Adadelta.

Adadelta is an extension of Adagrad, which can apply adaptive constraints on learning rate and simplify calculation. In contrast to Adagrad which sums all gradients squared, Adadelta only sums fixed-size terms and does not store them directly. The optimization of Adagrad is achieved only by approximate calculation of the average value.

6) RMSprop.

RMSprop can be counted as a special case of Adadelta. In fact, RMSprop still depends on the global learning rate, which is more suitable for processing non-stationary targets and performs better in RNN.

7) Adamo.

Adam is essentially RMSprop with momentum term, which mainly adjusts the learning rate of each parameter dynamically by using the first-order moment estimation and second-order moment estimation of gradient. Adam's advantage after offset calibration, in each iteration of the vector can be regulated to a range, and make the parameters more smoothly, and it combines the advantages of Adagrad and RMSprop can deal with sparse gradient and non-stationary target, for different parameters from adapt to challenge the learning rate,

optimize the overall performance of learning. Therefore, this network chooses Adam optimization algorithm. In the training process, this paper also tries to select various optimization methods, and finally Adam optimization algorithm is proved to be a better optimization algorithm.

3.6. Classification regression function

In logistic regression, the training set is composed of m marked samples c $\{(x^1, y^1),(x^2, y^2),\cdots,(x^m, y^m)\}$. Where the input feature $x^i \in \mathfrak{R}^{n+1}$. Logistic regression aims at dichotomous problems, and its function is defined as follows:

$$h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)} \qquad (2)$$

By training model parameter θ, the cost function can be minimized.

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}(y^i \log h_\theta(x^i)+(1-y^i)\log(1-h_\theta(x^i)))] \qquad (3)$$

In Softmax regression, the problem of multiple classification is solved. For a data set: $\{(x^1, y^1),(x^2, y^2),\cdots,(x^m, y^m)\}$ there is $y^i \in \{1,2,\cdots,k\}$. For a given input x, the softmax function is used to obtain the estimated probability value $p(y = j \mid x)$ for each category j. Therefore, it is possible to output an k-dimensional vector representing k estimated probability values. Suppose the function $h_\theta(x^i)$ is the following form:

$$h_\theta(x^i) = \begin{bmatrix} p(y^i=1 \mid x^i \mid \theta) \\ p(y^i=2 \mid x^i \mid \theta) \\ \vdots \\ p(y^i=k \mid x^i \mid \theta) \end{bmatrix} \qquad (4)$$

$$= \frac{1}{\sum_{j=1}^{k} e^{\theta_j T_x(i)}} \begin{bmatrix} e^{\theta_1 T_x(i)} \\ \vdots \\ e^{\theta_k T_x(i)} \end{bmatrix}$$

By minimizing the cost function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y^i = j\}\log\frac{e^{\theta_l T_x(i)}}{\sum_{j=1}^{k} e^{\theta_l T_x(i)}}]$$

(5)

Adding a weight attenuation term $\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2$.

Therefore, it can modify the cost function. The added attenuation term can penalize excessive parameter values. The cost function after the increase becomes:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y^i = j\}\log\frac{e^{\theta_l T_x(i)}}{\sum_{j=1}^{k}e^{\theta_l T_x(i)}}] \quad +$$

$$\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2 \qquad (6)$$

By adding this weight attenuation term, the cost function becomes strictly convex. At the same time, the Hessian matrix becomes an invertible matrix, and since the cost function is convex, algorithms such as gradient descent method can be used to ensure convergence to the local optimal solution.

3.7. Random inactivation

In the neural network, the typical training flow is to transmit the input forward through the network, and then transmit the error back through the cost function. Dropout is doing just that. The idea of Dropout is to set the inputs and outputs of hidden neurons to zero at a certain dropout ratio. If the selected hidden neuron "dropped out" is set to zero, it will not participate in the forward and backward propagation of the network. But the corresponding weight information will be retained. In the subsequent training of each input sample, the convolutional network adopts a different network structure from the previous training, but the weight of each training is shared. Dropout effectively reduce the occurrence of over-fitting and achieve regularization effect to some extent. In this paper, random inactivation is carried out at the full connection layer. In the last three fully connected layers, the neuron nodes are randomly inactivated with a dropout ratio of 0.5.

4. Experiments and analysis

In this paper, video frames of 5 kinds of basketball poses are extracted from KTH data set to construct a posture training database. The five postures are serving, passing, catching, running and shooting. The collected images are grouped into Pose1 ~ Pose5. Data sets are augmented by data enhancement (rotation, mirroring, random cropping, etc.). After the expansion, the data set contains about 2000 attitude data of each type, and a total of about 10,000 attitude images are used for network training and verification. After the data set classification is completed, 1/5 of the original sample of the data set is extracted as validation data.

4.1. Experimental environment

Under laboratory conditions, the experimental equipment required in this paper is a 64-bit Win10 system computer with Intel(R) Core (TM) CPU 3.50GHz, memory 64GB, and GeForce GTX 1060. The TensorFlow running environment is built based on the above hardware environment.

TensorFlow is a second-generation artificial intelligence learning system developed by Google based on DistBelief. TensorFlow uses data flow diagrams for calculations. Each node represents a mathematical operation in the diagram. The line edges in a data flow diagram represent a multidimensional array of nodes connected to each other, which is called a tensor. TensorFlow, as a general deep learning framework, is widely used in speech recognition, natural language processing, computer vision and other fields due to its flexible architecture, which allows computing to be carried out on a variety of platforms

It today supports Linux and Windows. Configure the TensorFlow operating environment, mainly relying on CUDA and OpenCV software environment. CUDA is a computing platform launched by NVIDIA. CUDNN is a GPU acceleration scheme designed specifically for Deep Learning framework. The software environment configured in this paper is CUDA8, CUDNN8 and TensorFlow-GPU version.

4.2. Specific implementation

Based on the network structure introduced above, the 11-layer convolutional network structure is shown in figure 3. The data set is convolved for 4 times, and pooled after each convolution. Finally, the full connection layer is entered. Feature map parameters generated by each layer of the network are shown in figure 3.
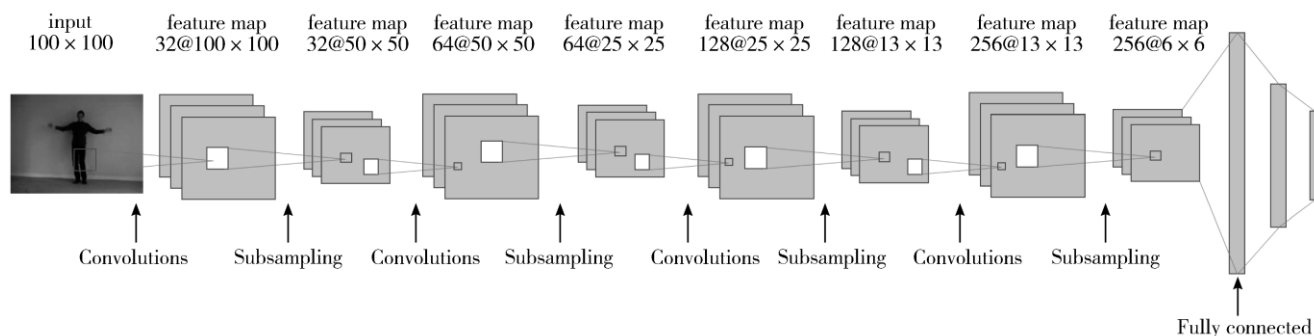
Figure 3. Network structure

In this paper, Batch-size and epoch are selected 64 times and 40 times to train the network model. The curve of training loss with training batches during training is shown in figure 4. As can be seen from figure 4, the network training model has an obvious initial downward trend of training loss with the increase of training batches. After the network iterations reaching 12 times, the training loss seems insignificant as it has dropped below 0.1, but the training loss is still decreasing according to the data. At the same time, this paper also records the variation curve of training data verification accuracy during the training process.
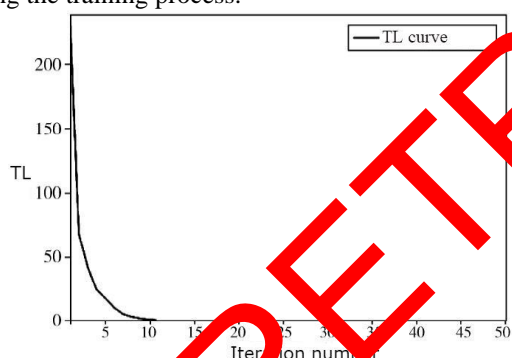

Figure 5. The curve of training accuracy changing with iteration times

Figure 4. Curve of training loss changing with the number of iterations (here TL denotes training loss)
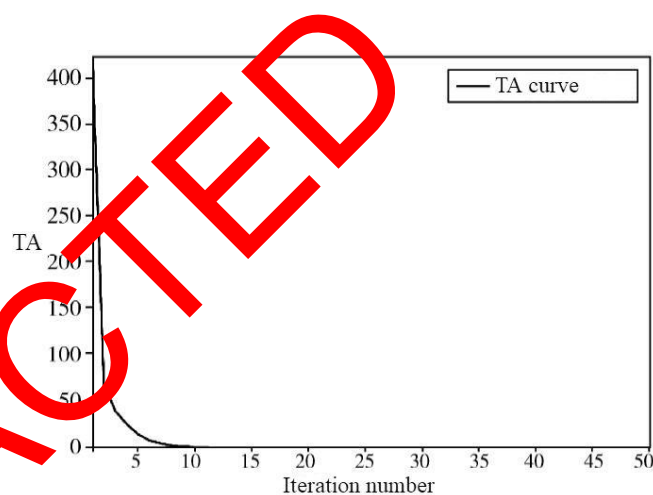
It can be seen from figures 4 and 5 that in the process of decreasing network training loss, the training accuracy also decreases with the number of iterations. The two curves fall by almost the same amount. This shows that with the increase of the number of training iterations, the recognition and classification performance of the network is gradually improved. The variation curve of validation accuracy with training batches is shown in figure 6. As can be seen from the figure, the validation accuracy increased significantly in the initial iteration. After the training iteration reached 33 times, the network tended to converge and the validation accuracy remained stable at 0.98218.
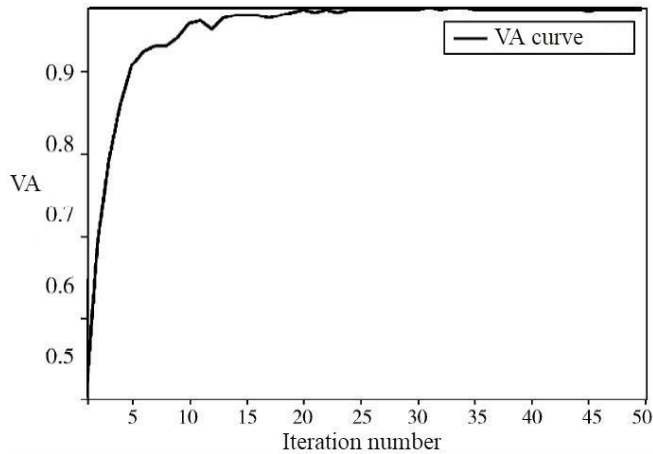
Figure 6. The validation accuracy changes with the number of iterations (VA denotes validation accuracy)

4.3. Results and analysis

In this experiment, five basketball poses are selected. The recognition result is obtained by training the network model.

It can be seen from figure 6 that the average recognition accuracy of the model is about 0.98 in the end, and the average recognition rate does not improve significantly after that. It can be seen that convolutional neural network has obvious advantages in basketball pose recognition, and the recognition effect is remarkable. Table 1 lists the recognition results based on random forest [23] and SVM [24]). Especially for attitude recognition results based on improved Gaussian kernel function [25], pose1 and Pose5 are compared, the recognition effects of Gaussian kernel function are 98% and 96%, while the recognition rates of Pose1 and Pose5 described in this paper are about 98.16% and 97.42%, indicating that the convolutional network model has a better performance in attitude recognition. As the data set and posture selected are different from other recognition methods, the comparison can only be made on the average recognition rate. Compared with machine learning methods such as random forest and SVM, it is not difficult to find that the recognition performance of the network in this paper is better.

Table 1. Comparison with different methods

Method	Average accuracy/%
Reference [23]	96.68
Reference [24]	97.94
Reference [25]	96.21
Proposed	98.23

5. Conclusion

This paper attempts to construct a convolutional neural network and HOG feature extraction recognition model for static human pose recognition. After repeated sample learning, the model can achieve better recognition effect on basic posture. Compared with SVM and other machine learning models, this model has better speed and generalization performance. Compared with traditional models, it does not need to design complex feature extraction methods, but the work of feature extraction is handed over to the network itself.

References

[1] F. Hu, Wang L, Wang S, et al. A Human Body Posture Recognition Algorithm Based on BP Neural Network for Wireless Body Area Networks[J]. China Communications, 2016, 13(8):198-208.

[2] Uriel H B, Victor A R. Real-Time Hand Posture Recognition for Human-Robot Interaction Tasks[J]. Sensors (Basel, Switzerland), 2016, 16(1).

[3] Dianhuai Shen, Xueying Jiang, Lin Teng. A novel Gauss-Laplace operator based on multi-scale convolution for dance motion image enhancement[J]. EAI Endorsed Transactions on Scalable Information Systems, 2021. http://dx.doi.org/10.4108/eai.17-12-2021.172439

[4] Xu W, Lee E J. A Novel Method for Hand Posture Recognition Based on Depth Information Descriptor[J]. Ksii Transactions on Internet & Information Systems, 2015, 9(2):763-774.

[5] Shoulin Yin, Hang Li, Asif Ali Laghari, et al. A Bagging Strategy-Based Kernel Extreme Learning Machine for Complex Network Intrusion Detection[J]. EAI Endorsed Transactions on Scalable Information Systems. 21(33), e8, 2021. http://dx.doi.org/10.4108/eai.6-10-2021.171247

[6] W. Ren, O. Ma, H. Ji and X. Liu, "Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches," in IEEE Access, vol. 8, pp. 135628-135639, 2020, doi: 10.1109/ACCESS.2020.3011697.

[7] Zhang S, Callaghan V. Real-time human posture recognition using an adaptive hybrid classifier[J].

International Journal of Machine Learning and Cybernetics, 2020:1-11.

[8] Ding W, Hu B, Liu H, et al. Human Posture Recognition Based on Multiple Features and Rule Learning[J]. International Journal of Machine Learning and Cybernetics, 2020, 11(11):2529–2540.

[9] Desheng Liu, Linna Shan, Lei Wang, Shoulin Yin, et al. P3OI-MELSH: Privacy Protection Point of Interest Recommendation Algorithm Based on Multi-exploring Locality Sensitive Hashing[J]. Frontiers in Neurorobotics, 2021. doi: 10.3389/fnbot.2021.660304.

[10] Wen J, Zhang N Z. Hand Posture Recognition Based on PCA[J]. Applied Mechanics and Materials, 2015, 738-739:631-634.

[11] Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.

[12] Ting-Ting Gao, Hang Li, and Shou-Lin Yin. Adaptive Convolutional Neural Network-based Information Fusion for Facial Expression Recognition [J]. International Journal of Electronics and Information Engineering. Vol. 13, No. 1, pp. 17-23, 2021.

[13] Chevtchenko S F, Vale R F, Macario V. Multi-Objective Optimization for Hand Posture Recognition[J]. Expert Systems with Applications, 2017, 92(feb.):170-181.

[14] Wan Q, Zhao H, Li J, et al. Hip Positioning and Sitting Posture Recognition Based on Human Sitting Pressure Image[J]. Sensors, 2021, 21(2):426.

[15] Shoulin Yin, Hang Li, Desheng Liu and Shahid Karim. Active Contour Modal Based on Density-oriented BIRCH Clustering Method for Medical Image Segmentation [J]. Multimedia Tools and Applications. Vol. 79, pp. 31049-31068, 2020.

[16] S. Yin and H. Li. HOT Region Selection Based on Selective Search and Modified Fuzzy C-Means in Remote Sensing Images[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 5862-5871, 2020, doi: 10.1109/JSTARS.2020.3025582.

[17] Shahid Karim, Ye Zhang, Shoulin Yin, Irfana Bibi. A Brief Review and Challenges of Object Detection in Optical Remote Sensing Imagery [J]. Multiagent and Grid Systems. 16(3), 227-243, 2020.

[18] Yin, S., Li, H. & Teng, L. Airport Detection Based on Improved Faster RCNN in Large Scale Remote Sensing Images [J]. Sensing and Imaging, vol. 21, 2020. https://doi.org/10.1007/s11220-020-00314-2

[19] Wei W, Liu Z, Huang L, et al. Predicting atypical visual saliency for autism spectrum disorder via scale-adaptive inception module and discriminative region enhancement loss[J]. Neurocomputing, 2020.

[20] Prasanna D, Prabhakar M. An effiecient human tracking system using Haar-like and hog feature extraction[J]. Cluster Computing, 2019.

[21] W. Zhou, S. Gao, L. Zhang and X. Lou, "Histogram of Oriented Gradients Feature Extraction From Raw Bayer Pattern Images," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 946-950, May 2020, doi: 10.1109/TCSII.2020.2980557.

[22] Xiaowei Wang, Shoulin Yin, Hang Li. A Network Intrusion Detection Method Based on Deep Multi-scale Convolutional Neural Network[J]. International Journal of Wireless Information Networks. 27(4), 503-517, 2020.

[23] Ding W, Hu B, Liu H, et al. Human Posture Recognition Based on Multiple Features and Rule Learning[J]. International Journal of Machine Learning and Cybernetics, 2020, 11(11):2529–2540.

[24] Athavale V A, Gupta S, D Kumar, et al. Human Action Recognition Using CNN-SVM Model[J]. Advances in Science and Technology, 2021, 105:282-290.

[25] Feng L, Liang H, Kang W, et al. Real-time hand posture recognition using hand geometric features and Fisher Vector[J]. Signal Processing Image Communication, 2019, 82(8):115729.