Research Article

An Improved Gesture Segmentation Method for Gesture Recognition Based on CNN and YCbCr

Yan Luo,¹ Gaoxiang Cui,² and Deguang Li ²

¹ChengDu Neusoft University, Chengdu 611844, China

²Luoyang Normal University, Luoyang 471934, China

Correspondence should be addressed to Deguang Li; lideguang.00@163.com

Received 6 May 2021; Revised 5 June 2021; Accepted 7 June 2021; Published 16 June 2021

Academic Editor: Yang Li

Copyright © 2021 Yan Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous improvement of people's requirements for interactive experience, gesture recognition is widely used as a basic human-computer interaction. However, due to the environment, light source, cover, and other factors, the diversity and complexity of gestures have a great impact on gesture recognition. In order to enhance the features of gesture recognition, firstly, the hand skin color is filtered through YCbCr color space to separate the gesture region to be recognized, and the Gaussian filter is used to process the noise of gesture edge; secondly, the morphological gray open operation is used to process the gesture data, the watershed algorithm based on marker is used to segment the gesture contour, and the eight-connected filling algorithm is used to enhance the gesture features; finally, the convolution neural network is used to recognize the gesture data set with fast convergence speed. The experimental results show that the proposed method can recognize all kinds of gestures quickly and accurately with an average recognition success rate of 96.46% and does not significantly increase the recognition time.

1. Introduction

With the development of the times and technology, human-computer interaction methods are gradually enriched. Typical interaction ways that are active in people's sight include face recognition, gesture recognition, micro-expression recognition, and human behavior recognition [1]. Human-computer interaction has evolved from the original computer as the main body to the person itself. Gesture is an intuitive and vivid way of behavior, containing a variety of rich and important information. In recent years, gesture recognition technology has been a research topic for researchers at home and abroad.

Gesture recognition can be studied from different fields, mainly including pattern recognition, signal processing, computer vision, and human-computer interaction, and typical methods include gesture recognition based on deep learning, gesture recognition based on Hidden Markov model (HMM) [2], gesture recognition based on geometric features, and gesture recognition based on computer vision [3]. The main problems faced by gesture recognition are

different regions, and different customs have different gesture meanings, which make gestures have diversity and polysemy [4, 5]. Gesture recognition based on data glove and optical marker makes gesture recognition restrictive and complex, and gesture recognition based on computer vision has many problems in hand gesture detection, such as hand gesture covering, changes of background color and scene, and various changes of image brightness [6], which cause great impact to gesture recognition and increase the difficulty of identification.

In the early stage of gesture recognition, gesture recognition mainly relies on gloves with sensors. Grime [7] first used marked gloves in 1983 to recognize simple gestures based on palm bone localization. In the next decade, gesture recognition technology using gloves as external devices has been developed. Takahashi and Kishino [8] successfully used gloves to recognize 46 common hand gestures. In view of the fact that gloves are not convenient enough, gloves with sensors have been replaced by finger marking methods. In the complex application background, static gestures can no longer meet the needs of reality, and gesture recognition in

dynamic videos has become the main research direction. In the research of hand segmentation in video data stream with complex background, Cui and Sun [9] used entropy analysis to segment the hand shape and then used the centroid of the hand shape to find the contour to recognize gestures, and this method has a high recognition rate. In the research of real-time detection, tracking and recognition of opponents in dynamic environment, Francke et al. [10] used boost classifier and skin color prior knowledge to track and proposed active learning and guidance method, and the method can achieve better results compared with boost algorithm, and in the simulation process, the detection rate of this method can be as high as 97%, the tracking rate can be as high as 99%, and the recognition rate is as high as 70%. Although the above algorithms and gesture recognition methods relying on peripherals can complete gesture recognition, they cannot meet the requirements due to the limitations of cost and peripherals. And now, gesture recognition based on deep learning has become a mainstream research topic [11–14].

Deep learning has unique capabilities in the field of computer vision, which imitates human brain's abstract memory of data to realize the machine's abstract expression of voice data, picture data, text data, and so on. Hubel and Wiesel [15] discovered that the visual cortex is hierarchical, and the high-level feature information is composed of the low-level features, and after multilevel extraction, the information obtained is less likely to be ambiguous, imitated by this feature. Kunihiro et al. [16] proposed a neural network model for a mechanism of visual pattern recognition in 1983. Based on these observations, Lecun et al. [17] proposed the CNN structure in their paper, and after continuous improvement, they proposed a multilayer neural network model called LeNet-5 to recognize handwritten digits. The network is not competent for training large-scale data, also the computer performance cannot meet the needs of large-scale storage, and then convolutional neural network has stagnated, while, with the improvement of computer performance and the development of big data, convolutional neural network has made a breakthrough in the number of layers, which makes it possible to train large amounts of data [18, 19]. In the 2012 image net image recognition contest, Alex et al. [20] proposed the AlexNet model and won the champion of the 2012 image recognition contest. The Relu activation function and dropout proposed in the network to prevent overfitting technology inspired the upsurge of convolutional neural networks, and then many researchers put forward more models with higher layers, such as ZFNet, VGGNet, GoogleNet, and ResNet. So far, various neural networks have appeared one after another and have been well applied. Zhang et al. [21] used deep learning for short-term voltage stability (STVS) evaluation of power systems, and their study established an evaluation model based on long and short-term memory (LSTM) by learning the time dependence of system dynamics after perturbation; the trained evaluation model is used to judge the stability of the system in real time, and this method can accurately and timely evaluate the stability of the system, which is better than traditional evaluation methods based on

shallow learning. Lin et al. [22] used deep learning to identify fatigue driving and got a high recognition rate, and many other applications such as [23–25]. In addition, many companies have invested a lot of energy in deep learning, which also promoted the development of deep learning.

Lee et al. [26] proposed a radar-based hand gesture recognition system to improve recognition accuracy, the domain adaptation is applied to hand gesture recognition to minimize the differences among users' gesture information in the learning and the use stage, and the classification accuracy of the proposed system was 98.8% on average for the target domain dataset. Pigou et al. [27] explored deep architectures for gesture recognition in video and proposed a new end-to-end trainable neural network architecture incorporating temporal convolutions and bidirectional recurrence. Since then, many researchers [28–31] study gesture recognition from different aspects, while the problem that the sample data is affected by environment, light, and other factors exists, as a result, the gesture features are not obvious, and the recognition rate of the neural network model is low.

In order to solve the problem that the gesture data is messy, and the gesture features are not obvious, the preprocessing scheme proposed in this paper can effectively extract the gesture features, and our contributions in this paper are as follows:

- (1) This research uses YCBCR color space skin color filter to segment the gesture area to enhance the features of the gesture.
- (2) Second, Gaussian filtering is used to process the noise around the segmented skin color area; then, we use the morphological opening operation to process the gesture image data, the watershed algorithm based on marker to segment the gesture area, and the eight-connected filling algorithm to fill the gesture area to enhance the gesture feature information.
- (3) Finally, the AlexNet convolutional neural network model is used to train the processed gesture data set. And the experimental results show that our average recognition success rate is 96.46% without significantly increasing the recognition time.

2. Gesture Data Preprocessing

The gesture data preprocessing is mainly to eliminate interference features of data pictures, highlight gesture features, simplify data scale, improve training efficiency, and increase recognition accuracy. The preprocessing methods in this paper include skin color detection, marker-based watershed algorithm, eight-connected seed filling algorithm, and scale normalization, and the preprocessing processes are shown in Figure 1.

2.1. Skin Color Gesture Detection Based on YCbCr. Gesture data is taken in different time periods, different lights, and different environments, and the gesture features are interfered by interference features, which increase the

FIGURE 1: Preprocessing processes of gesture recognition.

difficulty of gesture training. In order to reduce the brightness interference characteristics, the data picture in the RGB color space is converted to the YCbCr space, where the chroma and brightness can be separated, and the conversion relationship is

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}. \quad (1)$$

The Cr channel is extracted in the YCbCr color space. In order to reduce the interference of Gaussian noise on the gesture features, Gaussian filtering is performed on the data picture, and the two-dimensional Gaussian distribution is

$$f(x, y) = f(x)f(y) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}}. \quad (2)$$

In the formula, x and y are the coordinates of the data pixel, and σ_x, σ_y are parameters. The larger the value, the smaller the difference of Gaussian template coefficients, and the more obvious the smoothing effect of the picture. This paper uses a symmetric Gaussian kernel for filtering, and the skin color detection effect is shown in Figure 2.

2.2. Watershed Algorithm and Eight-Connected Seed Filling Algorithm. For the data samples processed by skin color detection, the data images are segmented by using the mark-based watershed algorithm, which is a way to connect the pixels of adjacent gray values into a contour to achieve the purpose of segmentation image, which is easy to segment images with noise and irregular gradient. The watershed algorithm based on mark solves the problem encountered by watershed algorithm, and the algorithm needs a mark image, which refers to a connected component of the image to be segmented, and the elevation of the connected component can be raised like a dam, which can prevent the local lower edge from being submerged and inseparable. The mark-based watershed algorithm can effectively segment the gesture features through the guidance of the marker graph, and the processing process of the mark-based watershed algorithm is shown in Figure 3.

After the gesture features are accurately segmented, the eight-connected seed filling algorithm is used to fill the internal pixels of the segmented irregular gesture area. The

FIGURE 2: Skin color detection.

eight-connected seed filling algorithm is the upgrade of the four-connected filling algorithm, which starts from an injection point in the area and extends in four directions, covering all pixels in the area, while the eight-connected seed filling algorithm accelerates the speed of filling the whole region by extending to eight directions. Through the eight-connected seed filling algorithm, the gesture feature data is obtained. The eight-connected seed filling algorithm is shown in Figure 4.

2.3. Scale Normalization and Gesture Labeling. After the segmentation and filling of the image data, the scale normalization operation can ensure the consistency of feature extraction and then label the gesture data with obvious features for neural network training. In this article, the image data obtained by segmentation and filling is normalized to $227 * 227$, which can effectively improve the convergence speed and the accuracy of the model in training and prevent the gradient explosion of the model.

3. Gesture Recognition Based on AlexNet Network Model

Deep learning refers to the use of deep neural network model for machine learning, through reasonable modeling, construction of reasonable loss function, forward propagation, backpropagation, and a large number of training data for machine training, so as to simulate the function from input data to output data and use the trained function to process new input, including but not limited to prediction and classification. Generally speaking, neural network is divided into three layers: input layer, hidden layer, and output layer. The input layer is a layer that receives input, and it receives a vector. Even if the input data is not a vector, for example, a picture needs to be transformed into a vector for input. The

FIGURE 3: Data image segmentation process.

FIGURE 4: Contour filling process.

hidden layer is the middle layer, whose function is feature transformation. The output layer is the layer of output results. The essence of neural network is a function, which completes the transformation from input data to output.

3.1. Analysis of Forward and Backward Propagation Process. The forward propagation of deep neural network means that, after the input data, the data flows into the next layer through calculation, and then to the output layer. The function between any two layers is

$$a_j^l = \sigma \left(\sum_{k=0}^n w_{jk}^l a_j^{l-1} + b_j^l \right), \quad (3)$$

where w_{jk}^l is the weight value of the k neurons in the $l-1$ layer to the j neurons in the l layer, b_j^l is the offset of the j neurons in the l layer, a_j^{l-1} is the j neuron in the $l-1$ layer, and a_j^l is the activation value obtained by forward propagation. In convolution neural network, convolution kernel

pooling is used for feature extraction, and the feature extraction of convolution layer is

$$F = f \left(\sum_{i=1}^m \text{con}(M_i, C_i) + B_i \right). \quad (4)$$

F is the output feature of the convolutional layer. If there are m convolutional layers, M_i is the i -th input feature, C_i corresponds to the convolution kernel, B_i is the bias matrix, and con is the convolution calculation function. After the convolutional layer, the output feature matrix is pooled to reduce the dimension of the feature matrix. After multiple convolutional pooling operations, the fully connected layer is used to activate the weight transformation, and the classification result can be obtained. The function expression is

$$F = f(WT + B_2). \quad (5)$$

In the formula, W is the weight matrix, T is the input characteristic data, and B_2 is the deviation. Use the cross

entropy (cross entropy) loss function to calculate the loss value of the forward propagation, and backpropagation adjusts the convolutional neural network parameters for the next iteration and minimizes the loss function output. The update function is as follows:

$$\begin{aligned}\omega_2 &= \omega_1 - \eta \nabla \omega, \\ b_2 &= b_1 - \eta \nabla b.\end{aligned}\quad (6)$$

In the formula, ω_2 is obtained by calculating and updating the learning rate and weight value of ω_1 , and b_2 is obtained by calculating and updating the learning rate and deviation of b_1 .

3.2. Model Building. This article refers to the use of the well-known convolutional neural network AlexNet, which was proposed in the 2012 Image net image recognition competition and won the championship. The Relu activation function and dropout prevention technology proposed in the network have good performance in neural network training. The AlexNet model has eight layers; among them, the first five layers are convolutional pooling layers, and the last three layers are fully connected layers. The softmax layer can output a 1000-dimensional matrix, that is, 1000 types of label distribution; the detailed structure of the model is shown in Table 1.

The input layer is $227 * 227$ image data, and the first layer of convolutional layer uses $11 * 11 * 3$ convolution kernel, using Relu activation function and then using $3 * 3$ step size 2 pooling unit pooling. The output of the pooling layer is a $27 * 27 * 96$ feature map, and then the output image is normalized for local response. The parameters are $K=2$, $n=5$, $\alpha = 10^{-4}$, $\beta = 0.75$, and the feature images with output of $27 * 27 * 96$ are divided into two groups, each of which is a $27 * 27 * 48$ feature image. In the second convolution layer, padding = 2, edge filling = 2, convolution kernel of $5 * 5 * 48$, and Relu activation function are used. Pooling is performed again, and local response normalization is performed after output. The parameters are the same as those of the first layer, and through the second convolutional layer, four groups of outputs are generated, each of which is a $13 * 13 * 128$ feature map.

The third and fourth layers have only convolutional layers, which use a $3 * 3 * 256$ convolution kernel, and they are filled with the edge of padding = 1; the convolution step is 1, and we use the Relu activation function. Through the convolution of the two layers, the output is a feature matrix with a size of $13 * 13 * 384$. The fifth layer uses padding = 1 for edge filling, a $3 * 3 * 192$ convolution kernel for convolution, and the output is a $13 * 13 * 256$ feature matrix. Use the Relu activation function, and then use a $3 * 3$ pooling unit with a step size of 2 for pooling, and the output is a $6 * 6 * 256$ feature matrix.

The sixth layer is fully connected, and dropout is used to prevent over fitting. The sixth layer is the first full connection layer, which has a total of 1024 convolution cores, each of which is $6 * 6 * 256$. Because the convolution core and input feature matrix are the same size, the convoluted pixel size is

$1024 * 1 * 1$, that is, the 1024 neurons. Dropout is used to suppress the overfitting. The seventh layer is the whole connecting layer, and the sixth layer, input as 1024 neurons, activates the function through Relu and then uses dropout to prevent over fitting. The eighth layer is the last output layer. The last 1024 neurons are classified by softmax function. Since this research only needs ten categories, the output is 10 float values. Through the above process, the AlexNet neural network model is established to train the preprocessed data.

3.3. The Optimization Strategy of Model. The AlexNet model uses the Relu activation function $\text{Relu} = \max(0, x)$ and the local response normalization (LRN-Local Response Normalization). LRN achieves local suppression to make the response larger and improves the generalization ability of the model. The function is

$$b_{x,y}^i = \frac{a_{x,y}^i}{\left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^j)^2\right)^\beta}, \quad (7)$$

where a represents the output of the i -th convolution kernel (x, y) coordinate in the feature graph through the Relu activation function, n represents the number of adjacent convolution kernels, and N represents the total number of convolution kernels in this layer. The values of k, n, α and β parameters are as follows:

$$\begin{aligned}k &= 2, \\ n &= 5, \\ \alpha &= 0.0001, \\ \beta &= 0.75.\end{aligned}\quad (8)$$

Dropout is used in the full connection layer of AlexNet model to effectively prevent over fitting. During training, neurons are randomly inactivated with a probability of 0.5, which doubles the convergence speed.

3.4. Training Process and Results. The preprocessed gesture data set is used as input to build the AlexNet network model, and the training of the model is the process of iterative forward propagation and backpropagation. With the parameter update of the hidden layer, the loss value is reduced, and the recognition accuracy of the model is improved. In this paper, the number of iterations is 3000, batch is set to 32, and the network model obtained through 3000 iterations of training can achieve a recognition accuracy of more than 95% on the test data set, and the change curve of recognition success rate in the training process is shown in Figure 5.

4. Experimental Results and Analysis

4.1. Experimental Environment. Our experiment is implemented on a PC equipped with Nvidia Geforce GT 635m (GPU), Intel Core i5-7500 processor, 8 GB memory, and Windows 10 system (64 bit). The identification and classification processes were implemented by PyCharm Community Edition 2020.3.2, configuration of tensorflow in

TABLE 1: Model structure.

Layer	The required action
Convolutional pooling layer	Convolution, activation, pooling, local response normalization
Convolutional pooling layer	Convolution, activation, pooling, local response normalization
Convolution layer	Convolution, activation
Convolution layer	Convolution, activation
Convolutional pooling layer	Convolution, activation, pooling
Fully connected layer	Activation, dropout
Fully connected layer	Activation, dropout
Output layer	Softmax classification function

FIGURE 5: Change curve of success rate.

Python3.5 for model training, installation of OpenCV package and PIL package for image processing, PyQt5 package to design an interactive interface.

4.2. Experimental Process and Results. The model trained in this experiment can detect ten types of gestures from 0–9, the data set is collected by computer, and each gesture data collects 200 image data. First, 20 gesture pieces of data are collected for each class of 10 subjects in different time periods, with a total of 2000 data samples. By training gestures 3, 4, and 5, the training effect is not good. Adding sample data can effectively improve the training effect. Therefore, each gesture category is added with 100 gesture data, and then the recognition success rate is significantly improved. The gesture representation method is shown in Figure 6.

In order to improve the training effect, the data set must be rich. For the same gesture, in different light sources, different angles, and different individuals, the position on the picture will be different, and Figure 7 shows different data of the same gesture.

As shown in Figure 8, some data of gesture training set are shown. The sample data is preprocessed as input and trained by the eight-layer neural network model, the trained neural network model has good recognition rate for gesture in complex background, and the recognition accuracy of all the test data set can reach more than 96%.

4.3. Analysis of Experimental Results. Compared with the Histogram of Oriented Gradients (HOG) and local binary patterns (LBP), the preprocessing method used in this paper improves the recognition rate of this algorithm compared with the other two methods, and the comparison results are shown in Table 2.

In the recognition rate test, 100 test samples were taken for measurement in different time periods, 92 times were

FIGURE 6: Gesture representation.

FIGURE 7: Different data for the same gesture.

successfully identified, 7 times of recognition errors were made, and gesture was not detected once. The results of the test in the extremely dark environment are not ideal, and the recognition errors of gestures 4 and 5 are 4 times. This is because the gesture features cannot be separated in the dark environment, and the similarity between gesture 4 and gesture 5 is too high. In normal environment, the model obtained by the preprocessing data method, which enhances the gesture feature, can achieve an extremely ideal

FIGURE 8: Partial gesture data sets.

TABLE 2: Comparison of recognition rate.

Method	Recognition rate
Algorithm	96.463
Local binary patterns (LBP)	86.734
Histogram of oriented gradients (HOG)	94.205

recognition rate. In order to test the robustness and stability of the model, four groups of sample data are randomly selected in the test data set, and each group of data is 30 test data graphs. Each group of data was identified and tested. After testing, the model has good robustness and stability. The comparison results are shown in Table 3.

Because dropout is used to prevent over fitting, the convergence speed of the model is improved. In the training process, after many attempts, improving the batch value can reduce the number of iterations and speed up the training speed. However, due to the reduction of the number of iterations, it may not be able to extract data features well and reduce the recognition rate. After testing, when the batch is 32, the training time and model stability are well unified. Figure 9 is the comparison chart of loss curves of different batch values.

In order to verify the recognition effect of the gesture recognition method on different individuals in this study, three people are randomly selected for testing in the laboratory. Each tester needs to test each gesture 50 times, divided into two groups, 25 times in each group; when there is sufficient light during the day, and when the light is insufficient at night, each of the 10 gestures is tested 500 times. The test results of the three tests are shown in Figure 10. The

TABLE 3: Recognition results.

Test data sample group number	Recognition success rate (%)
The first group	97.63
The second group	96.46
The third group	97.13
The fourth group	96.68

FIGURE 9: Comparison of loss value function curves of different Batch.

recognition rates of the ten gestures of the three testers are 94.4%, 94.6%, and 93.8%, respectively. The average success rate of gesture recognition by the three testers was 94.27%, and the recognition rate decreased slightly, proving that individual differences have a certain impact on gesture recognition. Among them, gesture 2 has the lowest average recognition rate, because gesture 2 and gesture 3 are too similar; gesture 1 and gesture 6 have the highest average

FIGURE 10: Recognition rate of different testers.

recognition rate, because gesture 1 and gesture 6 have obvious features and high recognition, and the results show that our gesture recognition method in this research has good scalability and compatibility.

5. Conclusion

In this paper, based on the deep learning of gesture recognition, by using skin color detection, marker-based watershed algorithm, and seed filling algorithm to preprocess the gesture data, the gesture data with obvious gesture characteristics can be obtained. Then, through the training of ten kinds of gesture data after preprocessing by AlexNet convolutional neural network, the recognition success rate of recognition test set data can reach 96.46% under natural light condition. The preprocessing method adopted in this paper can effectively reduce the interference of the environment on the gesture detection and recognition, and it does not bring longer training and detection time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Science and Technology Key Project of Henan Province under Grant no. 202102210370.

References

- [1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [2] D. Babu, M. Konyo, H. Nagano et al., "Stable haptic feedback generation for mid-air gesture interactions: a hidden Markov model-based motion synthesis approach," *ROBOMECH Journal*, vol. 6, no. 1, pp. 1–17, 2019.
- [3] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, Honolulu, HI, USA, July 2017.
- [4] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: a survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.
- [5] C. Szegedy et al., *Going Deeper with Convolutions*, presented at the CVPR, Boston, MA, USA, 2015.
- [6] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [7] G. J. Grimes, "Digital data entry glove interface device," *U.S. Patent*, No. 4,414,537, 1983.
- [8] T. Takahashi and F. Kishino, "Hand gesture coding based on experiments using a hand gesture interface device," *Acm Sigchi Bulletin*, vol. 23, no. 2, pp. 67–74, 1991.
- [9] J. Cui and Z. Sun, "Model-based visual hand posture tracking for guiding a dexterous robotic hand," *Optics Communications*, vol. 235, no. 4–6, pp. 311–318, 2004.
- [10] H. Francke, J. Ruiz-del-Solar, and R. Verschae, "Real-time hand gesture detection and recognition using boosted classifiers and active learning," in *Proceedings of the Pacific-Rim Symposium on Image and Video Technology*, pp. 533–547, Santiago, Chile, December 2007.
- [11] P. Molchanov, S. Gupta, K. Kim et al., "Hand gesture recognition with 3D convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–7, Boston, MA, USA, June 2015.
- [12] K. Xing, Z. Ding, S. Jiang et al., "Hand gesture recognition based on deep learning method," in *Proceedings of the 2018 IEEE 3rd International Conference on Data Science in Cyberspace (DSC)*, pp. 542–546, Guangzhou, China, June 2018.
- [13] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [14] S. Zhang, Z. Wei, J. Nie et al., "A review on human activity recognition using vision-based method," *Journal of Healthcare Engineering*, vol. 2017, Article ID 3090343, 31 pages, 2017.

- [15] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
- [16] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: a neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 826–834, 1983.
- [17] Y. LeCun, B. E. Boser, J. S. Denker et al., "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, pp. 396–404, Denver, CO, 1990.
- [18] X. X. Zhu, D. Tuia, L. Mou et al., "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.
- [19] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *CMC, Computers, Materials and Continua*, vol. 55, no. 1, pp. 1–16, 2018.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [21] M. Zhang, J. Li, Y. Li, and R. Xu, "Deep learning for short-term voltage stability assessment of power systems," *IEEE Access*, vol. 9, pp. 29711–29718, 2021.
- [22] Z. Lin, T. Qiu, P. Liu, L. Zhang, S. Zhang, and Z. Mu, "Fatigue driving recognition based on deep learning and graph neural network," *Biomedical Signal Processing and Control*, vol. 68, 2021.
- [23] Q. Su, T. Xu, and W. H. Zhou, "Predicting pore-water pressure in front of a TBM using a deep learning approach," *International Journal of Geomechanics*, vol. 21, no. 8, 2021.
- [24] N. Tomi, S. Sanna, S. Taavi et al., "Detecting pathological features and predicting fracture risk from dual-energy X-ray absorptiometry images using deep learning," *Bone Reports*, vol. 14, 2021.
- [25] M. Dujic, P. Luka, S. Marko, B. Marko, and L. Sven, "Automated Defect Detection from Ultrasonic Images Using Deep Learning," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2021.
- [26] H. R. Lee, J. Park, and Y.-J. Suh, "Improving classification accuracy of hand gesture recognition based on 60 GHz FMCW radar with deep learning domain adaptation," *Electronics*, vol. 9, no. 12, Article ID 2140, 2020.
- [27] L. Pigou, A. Van Den Oord, S. Dieleman et al., "Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, vol. 126, no. 2, pp. 430–439, 2018.
- [28] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University Computer and Information Sciences*, vol. 32, no. 4, pp. 1–11, 2019.
- [29] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2018.
- [30] M. Ciliberto, L. P. Cuspinera, and D. Roggen, "WLCSSLearn: learning algorithm for template matchingbased gesture recognition systems," in *Proceedings of the 2019 Joint 8th International Conference on Informatics, Electronics and Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision and Pattern Recognition (icIVPR)*, pp. 91–96, IEEE, Spokane, WA, USA, June 2019.
- [31] F. Mueller, F. Bernard, O. Sotnychenko et al., "Generated hands for real-time 3d hand tracking from monocular rgb," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–59, Salt Lake City, UT, USA, June 2019.