Hindawi
Advances in Multimedia
Volume 2021, Article ID 6654561, 9 pages
https://doi.org/10.1155/2021/6654561

Research Article

Face Alignment Algorithm Based on an Improved Cascaded Convolutional Neural Network

Xun Duan, Yuanshun Wang, and Yun Wu

School of Computer Science and Technology, GuiZhou University, GuiYang 550025, China

Correspondence should be addressed to Yuanshun Wang; yswang837@gmail.com

Received 1 November 2020; Revised 12 February 2021; Accepted 30 March 2021; Published 8 April 2021

Academic Editor: Deepu Rajan

Copyright © 2021 Xun Duan 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.

Aiming at the problem of a large number of parameters and high time complexity caused by the current deep convolutional neural network models, an improved face alignment algorithm of a cascaded convolutional neural network (CCNN) is proposed from the network structure, random perturbation factor (shake), and data scale. The algorithm steps are as follows: 3 groups of lightweight CNNs are designed; the first group takes facial images with face frame as input, trains 3 CNNs in parallel, and weighted outputs the facial images with 5 facial key points (anchor points). Then, the anchor points and 2 different windows with a shake mechanism are used to crop out 10 partial images of human faces. The networks in the second group train 10 CNNs in parallel and every 2 networks' weighted average and colocated a key point. Based on the second group of networks, the third group designed a smaller shake mechanism and windows, to achieve more fine-tuning. When training the network, the idea of parallel within groups and serial between groups is adopted. Experiments show that, on the LFPW face dataset, the improved CCNN in this paper is superior to any other algorithm of the same type in positioning speed, algorithm parameter amount, and test error.

1. Introduction

The task of face alignment [1–3] is also called facial landmark localization. In the case of a given facial image, the algorithm locates the key points of the face, including the left eye, the right eye, the tip of the nose, and the left and right corners of the mouth. The difficulty lies in quickly and accurately predicting the coordinate values of 5 key points of a given facial image. Facial landmark localization [4, 5] is a relatively core algorithm business that plays a crucial role in many scientific research and application topics. It has always been a research issue in the fields of image processing, pattern recognition, and computer vision.

The current facial landmark localization methods are mainly divided into 3 categories: AAM [6] (active appearance model) based on traditional methods and ASM [7] (active shape model) based on models and methods based on deep learning. The AAM method and model-based ASM extract the semantic features of a given facial image by iteratively solving an optimization problem [8] under various constraints. In addition to the high computational complexity, it also easily falls into a local minimum; with the rise of deep learning methods, face alignment tasks have achieved extremely accurate results in terms of accuracy. However, the current high-precision detection algorithms mostly use deep networks (such as VGG [9] and Resnet-34 [10]), which creates a substantial number of parameters, resulting in the time and space complexity increasing dramatically, and the chain rule can easily lead to gradient vanishing and gradient explosion in the deep network [11], which can easily cause the network to fail to learn useful rules. For this reason, this paper improved the cascaded convolutional neural network [12, 13] from the aspect of network structure, raising the random perturbation factor (shake) and the data scale. Under the condition of ensuring accuracy, lightweight [14] convolutional neural networks are designed to reduce the number of parameters that the networks need to learn and accelerate the speed at locating key points on the face.

2. Related Theories

Convolutional neural networks (CNNs) [15] are a new type of artificial neural network proposed by combining traditional artificial neural networks and deep learning. Since fully connected neural networks have a large number of parameters when processing image data, CNNs optimize the structure of the traditional fully connected neural networks by introducing weight sharing [16] and local perception methods, which greatly reduces the number of learnable parameters of the fully connected neural networks. Since the feature maps output by the CNN pooling layers have the advantages of rotation invariance and translation invariance, CNNs have natural robustness [17] for processing image data. In the common CNN model architecture, most CNNs use the convolutional layer and the pooling layer alternately to extract the semantic information of the facial image from low to high. After the channel numbers and size of the feature maps reach a certain dimension, the feature maps are arranged in order and converted into a one-dimensional feature vector, which is connected with the fully connected layer for dimensional transformation. The operation process of the convolutional layer can be expressed as follows:

$$X^{(l,k)} = f\left(\sum_{p=1}^{n_l-1}\left(W^{(l,k,p)} \otimes X^{(l-1,p)}\right) + b^{(l,k)}\right). \qquad (1)$$

In formula (1), $X^{(l,k)}$ represents the k-th feature map output by the l-th hidden layer, nl represents the number of channels of the l-th layer feature map, and $W^{(l,k,p)}$ represents the convolution kernel used when mapping the p-th group of feature maps in the $(l-1)$-th layer to the k-th feature maps in the l-th layer. The algorithm in this paper uses max-pooling. After the pooling operation, the size of the feature map is reduced to one-fold of the original step size. Max-pooling can be expressed as follows:

$$X^{(l+1,k)}(m,n) = \max_{0<a,b<s}\left\{X^{(l,k)}(m \cdot step + a, n \cdot step + b)\right\}. \qquad (2)$$

In formula (2), $X^{(l+1,k)}$ (m, n) represents the value of the k-th group of feature map coordinates output from the $(l+1)$-th layer at (m, n).

When optimizing the CNN model, the back propagation method is used to update all the connection weights between neurons to minimize the loss function. Since the face alignment task is a regression task [18], the mean square error (MSE) is used to define its loss function, which can be expressed as follows:

$$\text{MSE} = \frac{1}{2N}\sum_{i=1}^{N}\left\|O_i - P_i\right\|_2^2. \qquad (3)$$

In formula (3), N is the number of nodes in the input layer of the neural network, O is the artificially labeled value, and P is the predicted value of the neural networks.

3. Improved CCNN

To improve the efficiency and accuracy of the algorithm detection in this paper, the CCNN network is improved in the following 4 aspects:

Improvement 1: designing shallower networks: deep networks create a large number of parameters, resulting in a sharp increase in time and space complexity. Additionally, the chain rule can easily lead to gradient vanishing and gradient explosion in deep networks, and it easily causes the networks to fail to learn useful rules. The CCNN algorithm has a total of 23 CNNs, and each CNN contains a maximum of 8 layers and a minimum of 6 layers. The unnecessary convolution and pooling layers are removed, and the number of algorithm parameters is greatly reduced to ensure the running speed of the algorithm.

Improvement 2: a shake mechanism is proposed to solve the problem that the rules learned by the networks tend to the geometric center of the windows, which improves the generalization ability (robustness) of the model. The positioning accuracy and speed are increased by designing the size of windows.

Improvement 3: adopting multilevel cascaded recursive CNNs: it is verified by experiments that the 3-stage cascaded structure of CCNN is selected in this paper, which can guarantee a high positioning accuracy within an acceptable time range.

Improvement 4: designing a smaller input image size: a smaller input image size can speed up the training of the networks, and the number of channels of the convolution kernel can be increased to compensate for the information loss caused by a smaller input image.

3.1. CCNN Framework Design.

The framework of the improved cascaded convolutional neural network in this paper is divided into 3 stages, which realizes the process from global coarse positioning to local precise positioning.

The overall design idea of the CCNN algorithm adopts the recursive idea, and the recursive termination condition is "the number of network stages that balance positioning accuracy and positioning time complexity." Based on this, a cascaded convolutional neural network with 3 stages is designed; in addition to guaranteeing a high positioning accuracy rate, it is also necessary to ensure that the positioning results are obtained within an acceptable time frame. As shown in Table 1, in the design of the CCNN algorithm, each stage recursively generates the key points of the face.

Table 1: Design of the CCNN algorithm.

Stage 1	Input: X (face images with face frames.)
	Step 1: resize X to (39*39*3)
	Step 2: after 2 sets of convolutional, pooling layer, and 2 fully connected layers, generate a candidate set of key points
	Step 3: same as step 2; generate candidate face key point coordinates
	Output: y1 (face image with 5 weighted key points)
Stage 2	Input: use 2 different windows with shake to crop y1 to obtain 10 partial face images
	Step 1: similar to stage1-step2; generate a candidate set of key points
	Step 2: every 2 CNNs colocate a key point
	Output: y2 (face image with 5 weighted key points)
Stage 3	Input: use 2 smaller different windows with shake to crop y2 to obtain 10 partial face images
	Step 1: similar to stage2-step1; generate a candidate set of key points
	Step 2: every 2 CNNs colocate a key point
	Output: y3 (face image with 5 weighted key points)

3.1.1. The First Stage. As shown in Figure 1, the original image is marked with the initial face frame, and Figure 1(a) is cropped into 3 areas according to the face distribution of the LFPW dataset (respectively, the key point area contained in the initial face frame, left, right eye, and nose tip key points area, and nose tip, left, and right mouth corner key points area); these 3 areas are used as the input of the 3 CNNs in the first stage, and the coordinate values of 5 key points are weighted average output to obtain Figure 1(b).

The rough positioning of global key points is realized. The weighted average method is shown as follows:

$$(x, y)_p = \frac{1}{N}\sum_{i=1}^{N}(\widehat{x}_i, \widehat{y}_i). \tag{4}$$

In formula (4), P belongs to the set of {left eye, right eye, nose tip, left mouth corner, right mouth corner}, and N is the number of repeated positions.

3.1.2. The Second Stage. The 5 key points output in the first stage are used as anchor points and expanded to the top, bottom, left, and right of the anchor points to obtain the current 10 windows. The expansion method is based on the following formula:

$$\begin{cases} \text{windows}_i = (a_1{}^*h, b_1{}^*w), & i = 1, 2, \ldots, 5, \\ \text{windows}_j = (a_2{}^*h, b_2{}^*w), & j = 6, 7, \ldots, 10. \end{cases} \tag{5}$$

In formula (5), windows$_i$ represents the 5 windows of the first group, window$_j$ represents the 5 windows of the second group, a and b are hyperparameters, a1 and b1 take the value of 0.16; a2 and b2 take the value of 0.18, and h and w are the height and width of the initial face frame. Then, according to the size of the windows and shake (shake in the paper follows a normal distribution, namely, shake ~ N (0, 1e – 4), as arrow ①), 10 partial human face images are to cut out with a random perturbation factor to obtain Figure 1(c). Shake can be regarded as a kind of data augmentation operation. Its essence is a type of translation strategy. It can shift the window randomly up, down, left, and right by shake units. These 10 partial facial images are taken as the input of the 10 CNNs in the second stage, and every 2 CNNs use formula (4) to jointly weight and average a key point to obtain Figure 1(d), thereby realizing local key point positioning.

3.1.3. The Third Stage. Taking the 5 key points output in the second stage as anchor points and by designing smaller windows and shake (a1 and b1 take the value of 0.08; a2 and b2 take the value of 0.09, shake is the same as the second stage, as arrow ②) to cut out the smaller 10 partial images of the facial key points, we obtain Figure 1(e). These 10 partial images are used as the input of the 10 CNNs in the third stage to obtain Figure 1(f), and every 2 CNNs are jointly weighted averaged by formula (4) to locate a key point to obtain the accurate localization of the partial image of the face, which is the final output of the algorithm in this paper.

3.2. Network Structure Design. Since the algorithm in this paper is based on the recursive idea, now we take the second stage (level 2), the LE21 CNN and LE22 CNN, which locate the key point of the left eye (LE), as examples to describe the network structure design of the CCNN (F is the full face, L is left, R is right, E is the eye, N is the nose, M is the corner mouth, and LE21 CNN represents the first CNN that locates the left eye in the second stage, the same as follows).

3.2.1. Level 2 Network Structure. Figure 2 shows the level 2 network structure. This stage contains 10 lightweight CNNs, namely, LE21, LE22, RE21, RE22, N21, N22, and LM21.

LM22, RM21, and RM22: every 2 CNNs are jointly weighted average by formula (4) to locate a key point and combined with the positioning results of the other 8 CNNs, and finally, the level 2 output is obtained.

3.2.2. Level2-LE21 CNN Network Structure. To improve the positioning speed and efficiency of the algorithm in this article, we design a new cascaded convolutional neural network in this section, which accelerates the training and testing of the network by reducing the number of layers of each CNN and reducing the size of the input image. Compared with reference [13], this paper reduces the number of network layers from 12 to 6 layers, and the input size of the second stage is reduced from 32*32* 3 to 15*15*3.

Figure 3 shows the network structure of level2-LE21 CNN. We input 15*15*3 left eye local area map, after 2 sets of convolutional pooling layers and 2 fully connected layers, we output the left eye prediction coordinates value of the

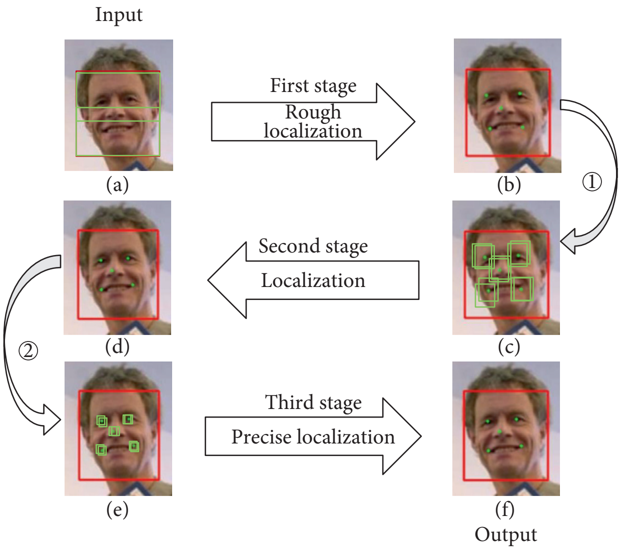

Figure 1: CCNN algorithm framework.

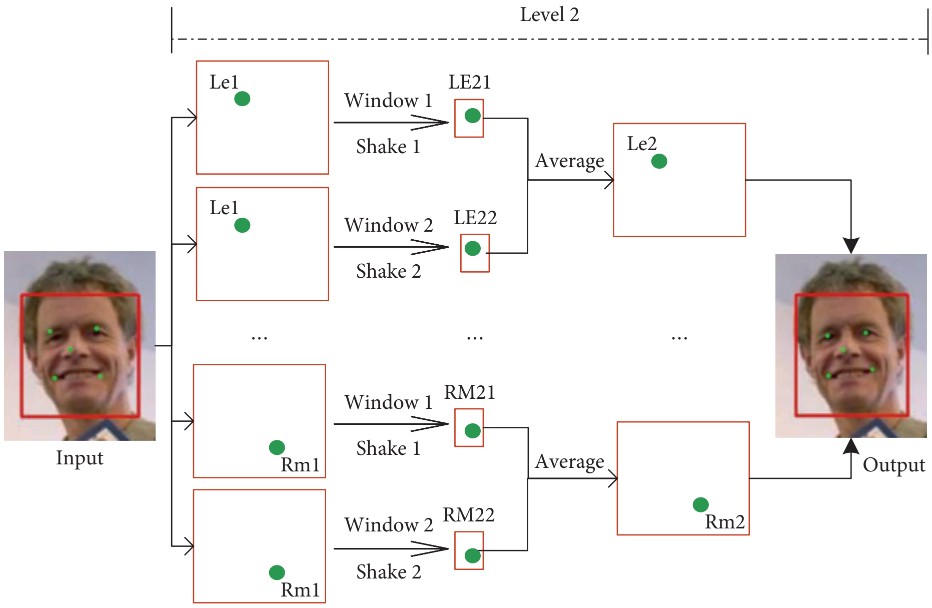

Figure 2: Schematic diagram of the level 2 network structure.

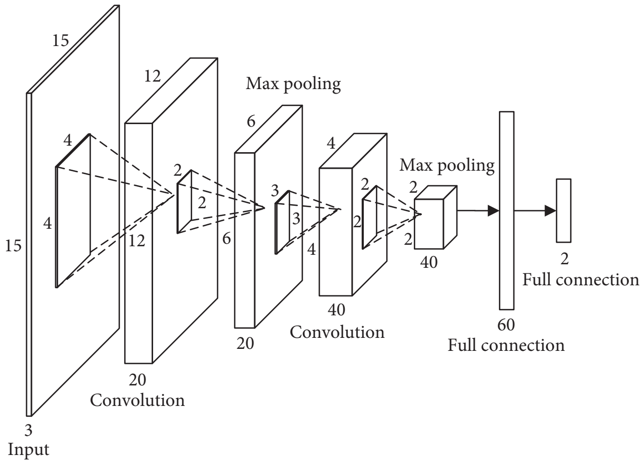

Figure 3: Network structure of level2-le21 CNN.

LE21 CNN, combined with the left eye coordinate value output by the LE22 CNN, and finally, we weighted average the left eye coordinate value of level 2 in the manner of formula (4).

3.2.3. Proposing the Shake Mechanism and Windows. The entire algorithm flow is carried out recursively in stages. As shown in Figure 1, the original red frame of the input picture in the first stage can be regarded as a larger window, and each small green frame of the input picture in the second stage is a smaller window. Since the positioning method of the algorithm in this paper uses the relative positioning of windows, the more accurate the windows, the more accurate the model positioning; the smaller the windows, the faster the model positioning speed. To avoid the immutability of windows, the shake mechanism is proposed to slightly shake windows, that is, translation (which has been explained in detail in the second stage of Section 3.1.2); it can also be regarded as a method of data augmentation, thereby improving the generalization ability of the model.

3.2.4. Model Parameters. The network structure parameters of the first stage are shown in Table 2, where the fully connected layers of F1, EN1, and NM1 are 10d-fc2, 6d-fc2, and 6d-fc2, respectively. The network structure of the third stage is the same as that of the second stage. The total number of network parameters is only 643,920, which is approximately 2.46 MB.

3.3. CCNN Model Training. The model training method in this paper adopts the idea of parallel within groups and serial between groups. For each lightweight CNN, a rectified linear unit (ReLU) is used after the convolutional layer to increase the nonlinear expression ability of the model, max-pooling layers are used to downsample the feature map of the convolutional layers, and the fully connected layers use dropout technology to improve the robustness of the model (the value is 0.25). The initial learning rate is $5e-3$, and every 100 iterations, the learning rate decays to 90% of the original (the other hyperparameters are selected according to the actual situation, such as batch size and iteration times). The stochastic gradient descent (SGD) algorithm is used to update the weights of all connections between neurons during network training. To additionally increase the robustness of the model, L2 regularization is introduced to penalize the learnable parameter W, and then, the final expression of the newly designed loss function is as follows:

$$E = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{2N}\sum_{i=1}^{N}\left\|y_i^j - d_i^j\right\|_2^2\right) + \frac{1}{2}\eta\|W\|_2^2. \tag{6}$$

In formula (6), m is the batch size and W is the weight matrix in the network. This weight matrix W is updated during the process of error back propagation. Before the network starts training, the Xavier [19] method is used to initialize the weight matrix W0. The weight matrix Wt + 1 updated after $t+1$ iterations can be expressed as follows:

$$W_{t+1} = W_t - \lambda \cdot \frac{\partial E}{\partial W_t}. \tag{7}$$

4. Experiments and Results

The operating system used in the experiment is Centos7 64 bit, a Dell server equipped with 2 RTX2080TI graphics cards and 32 GB memory, and the code running environment is the PyTorch framework.

4.1. Dataset. These experiments use the facial dataset LFPW, a total of 13,466 facial images. Each facial image has 3 channels with RGB and has relevant coordinate annotation information (including key point coordinates and initial face frame coordinates). The original data use 10,000 images as the training set and 3,466 images as the test set. Data augmentation [20, 21] can be used to increase the number of training samples to effectively improve the performance of the convolutional neural network. To be consistent with the actual facts, we set the left 15-degree and right 12-degree center rotation (the label coordinates should also be rotated), and the rotation method is given in formula (8), where θ is the rotation angle.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{8}$$

Then, the training dataset is flipped horizontally (the label coordinates should also be flipped accordingly), and 60,000 training data are finally obtained. Each training data requires data standardization (minus the mean and dividing the variance). In addition, for each training data, it is needed to eliminate the impact of the length and width of the face frame (windows at each stage). The elimination method is shown as follows:

$$\begin{cases} x\ \text{direction:} & \dfrac{|x' - x|}{w}, \\ y\ \text{direction:} & \dfrac{|y' - y|}{h}. \end{cases} \tag{9}$$

In formula (9), x and y are the horizontal and vertical coordinates of the upper left corner of the windows, x' and y' are the horizontal and vertical coordinates of the key points predicted by the current windows, and w and h are the width and height of the windows.

4.2. Experiment Results

Experiment 1. Verifying the necessity of the 3-stage cascaded structure.

As shown in Figure 4, the vertical axis is the test error, the horizontal axis is the facial key points, and the blue, red, and green broken lines are the test errors of the first stage, the second stage, and the third stage, respectively.

Table 2: Network structure table of the first stage.

Name	Input	Convolution kernels/steps	Output	Parameters
conv1	39*39*3	4*4* 20/1	36*3 6* 20	960
pool1	36*36*20	2* 2/2	18*18*20	-
conv2	18*18*20	3*3* 40/1	16*16*40	7200
pool2	16*16*40	2* 2/2	8*8*40	-
conv3	8*8*40	3*3* 60/1	6*6*60	21600
pool3	6*6*60	2* 2/2	3*3*60	-
120d-fc1 64800				
10d-fc2 (F1) 6d-fc2 (EN1) 6d-fc2 (NM1)				

The test error of the second stage exceeded less than that of the first stage. The test error of the third stage also decreased compared with the second stage; the average test error decreased from 2.21% to 1.37% and then to 1.03%. To balance the positioning accuracy and positioning speed comprehensively, this article does not cascade the fourth stage of CNNs.

Experiment 2. Verifying the necessity of under windows.

Figure 5 shows the impact of the presence or absence of window frames on the test error of each key point in the first stage. The vertical axis is the test error, the horizontal axis is the facial key points, and the blue broken line is windows-, which uses absolute positioning; the red broken line indicates the relative positioning of windows+.

The relative positioning method is more stable, and the test error at each key point is lower than that of the absolute positioning method. The average test error decreased from 3.38% to 2.35%. Because the first stage uses global key point coarse positioning of the first stage, the test errors of the blue and red broken lines are both high.

Experiment 3. Verifying the necessity of the under shake mechanism.

Figure 6 shows the impact of the presence or absence of the shake factor on the test error of each key point in the first stage.

The vertical axis is the test error, and the horizontal axis is the facial key points. The blue broken line is shake-, which means no data augmentation operation; the red broken line is shake+, which denotes data augmentation on the partial image. The test error of the red broken line, after data augmentation, is lower than that of the blue broken line, which shows that the shake mechanism has improved the generalization ability of the model. The average test error dropped from 2.35% to 1.90%. Because the first stage uses global key point coarse positioning, the test errors of the blue and red broken lines are both high.

Experiment 4. Comparison with other algorithms of the same type in LE error, RE error, N error, LM error, RM error, and average test error.

Figure 7 shows the test error of each key point of each algorithm. The vertical axis is the test error, and the horizontal axis is the facial key points.

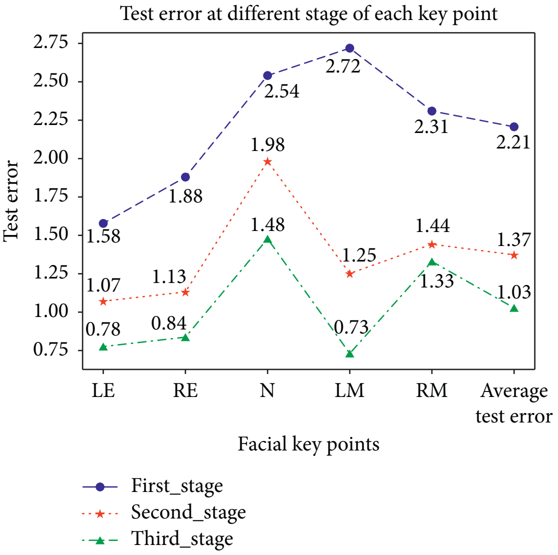

Figure 4: Test error graph of key points in different stages.

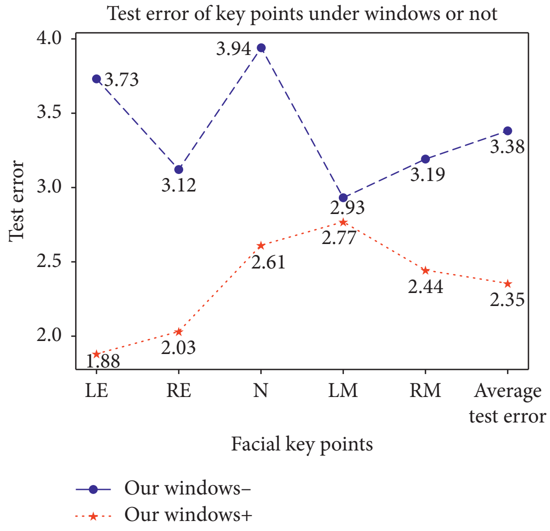

Figure 5: Test error graph of key points under windows or not.

The blue and red broken lines are the algorithms of Sun [22] and Chen Rui [13], respectively, and the green broken line is the algorithm in this paper; it can be seen that the algorithm in this paper is lower than any other algorithm of the same type in LE, RE, N, LM, and average test error, and the average test error reaches 1.03%. The test error at the key points of RM is slightly higher than that of Chen Rui's [13] algorithm, which indirectly

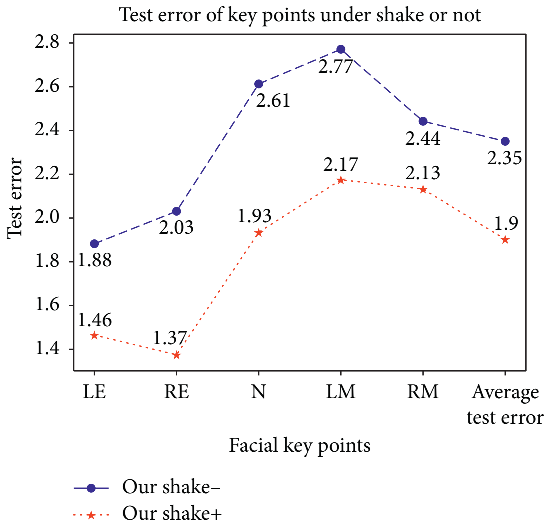

FIGURE 6: Test error graph of key points under shake or not.

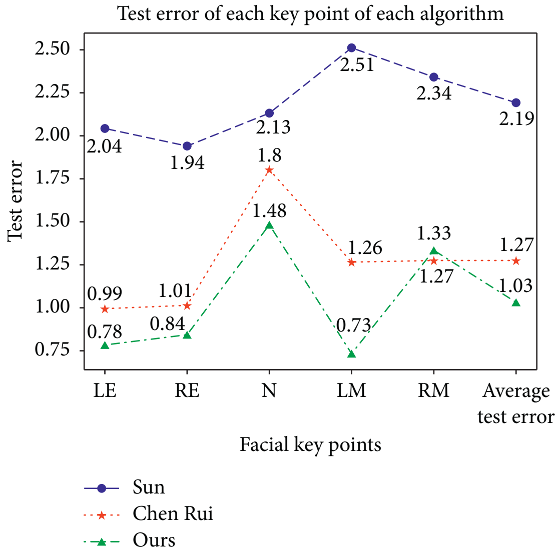

FIGURE 7: Test error graph of key points of each algorithm.

indicates that each network needs separate hyperparameters to train.

In addition, the learnable parameters in this algorithm are only 643920, approximately 2.46 MB, which is approximately 58% of the parameters of Chen Rui [13] and 5.6% of the ResNet18. It only requires 14.7 milliseconds to process a facial image on the GPU. As shown in Table 3, it greatly reduces the consumption of hardware resources and can be used as an embedded algorithm for cameras or apps used in mobile devices such as mobile phones.

Figure 8 shows the effect picture of the facial landmark localization of the algorithm in this paper. The first line of facial images is output 1 of the first stage of CNN, and so on; it can be seen that the output of the first CNN stage (global key points coarse positioning stage) has flaws, and the positioning accuracy of the second CNN stage (local key points positioning stage) has improved significantly. The positioning accuracy of the third CNN stage (accurate positioning of local key points) also improved and was no longer distinguishable by the naked eye.

TABLE 3: Parameters and speed of each algorithm.

Methods/indicators	Parameters (MB)	Speed (ms)
Algorithm of this paper	**2.46**	**14.7**
ResNet18	44.59	-
ResNet34	85.15	-
Chen Rui [13]	4.24	15.9

Under the influence of distorted expressions and diverse postures, the algorithm still achieves accurate positioning. If we face tasks with very high real-time requirements, then only the first 2 layers of cascaded convolutional neural networks are sufficient.

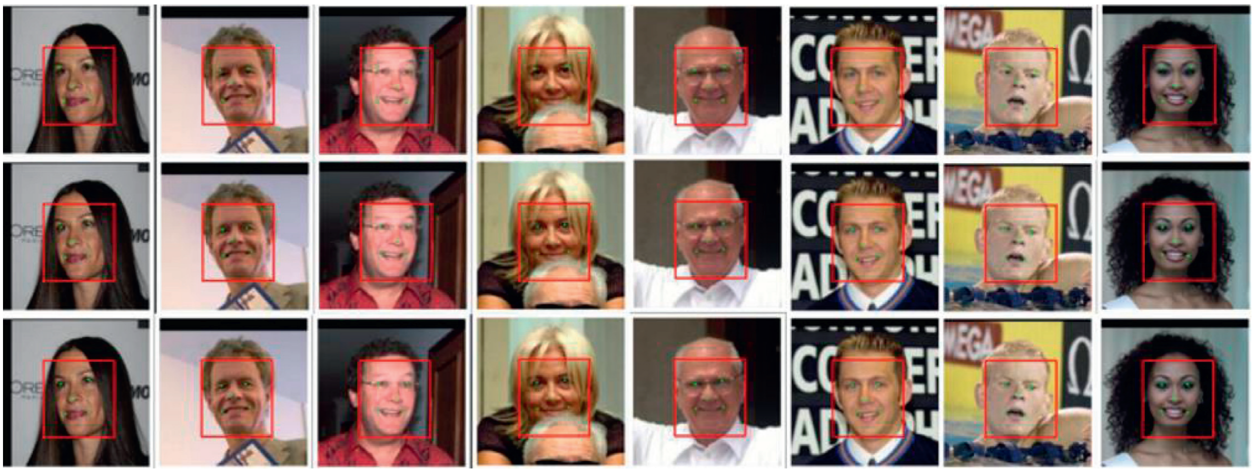

Figure 8: Effect picture of facial landmark localization.

5. Conclusions

The improved face alignment algorithm based on a cascaded convolutional neural network in this paper realizes the process from coarse positioning to precise positioning by designing a 3-stage network structure. The improvement of windows accelerates the training of the network and improved the positioning accuracy rate of the key points. The proposed shake factor can be regarded as a data augmentation operation, which improves the generalization ability of the model. The test on the LFPW face dataset shows that the average test error of the algorithm in this paper reaches 1.03%, which is approximately 1.16% lower than that of the same type of algorithms; in addition, it only takes 14.7 milliseconds to process a facial image on the GPU, which is 1.2 faster than Chen Rui's [13] in milliseconds, and a small-scale and efficient facial landmark localization network is realized. The algorithm shows good antiinterference ability with gestures and expressions. Future work can perform further research on data augmentation and add detection technology to give a more accurate initial face frame.

Data Availability

The data used to support the findings of the 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 by the National Natural Science Foundation of China ([2018] 61741124) and Guizhou Science and Technology Plan Project ([2018] 5781).

References

[1] G. Q. Qi, J. M. Yao, H. L. Hu et al., "Face landmark detection network with multi-scale feature map fusion," *Application Research of Computers*, vol. 1-6, pp. 1001–3695, 2019.

[2] J. Xu, W. J. Tian, and Y. Y. Fan, "Simulation of face key point recognition and location method based on deep learning," *Computer Simulation*, vol. 37, no. 06, pp. 434–438, 2020.

[3] J. Li, *The Research and Implement of Face Detection and Face Alignment in the Wild*, NanJing University, Nanjing, China, 2019.

[4] X. F. Yang, *Research on Facial Landmark Localization Based on Deep Learning*, Xiamen University, Xiamen, China, 2019.

[5] C. X. Jing, *Research on Face Detection and Face Alignment Based on Deep Learning*, China Jiliang University, Hangzhou, China, 2018.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper et al., "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, 1995.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proceedings of the 5th European Conference on Computer Vision-Volume II-Volume II*, Springer, Berlin, Heidelberg, June 1998.

[8] X. Yu, C. Ma, Y. Hu et al., "New neural network method for solving nonsmooth pseudoconvex optimization problems," *Computer Science*, vol. 46, no. 11, pp. 228–234, 2019.

[9] A. Sengupta, Y. Ye, R. Wang et al., "Going deeper in spiking neural networks: VGG and residual rrchitectures," *Frontiers in Neuroence*, vol. 13, 2018.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke et al., "Inception-v4, inception-resNet and the impact of residual connections on learning," 2016, https://arxiv.org/abs/1602.07261.

[11] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," 2015, https://arxiv.org/abs/1512.03385.

[12] J. H. Xie and Y. S. Zhang, "Caspn: cascaded spatial pyramid network for face landmark localization," *Application Research of Computers*, vol. 1-6, 2020.

[13] R. Chen and D. Lin, "Face key point location based on cascaded convolutional neural network," *Journal of SiChuan Technology University (natural Edition)*, vol. 30, pp. 32–37, 2017.

[14] L. H. Xu, Z. Li, J. J. Jiang et al., "A high- precision and lightweight facial landmark detection algorithm," *Laser & Optoelectronics Progress*, vol. 1-12, 2020.

[15] J. D. Lin, X. Y. Wu, Y. Chai et al., "Structure optimization of convolutional neural networks:a survey," *Journal of Automatica Sinica*, vol. 46, no. 01, pp. 24–37, 2020.

[16] F. Y. Hu, L. Y. Li, X. R. Shang et al., "A review of object detection algorithms based on convolutional neural

networks," *Journal of SuZhou University of Science and Technology (Natural Science Edition)*, vol. 37, no. 02, pp. 1–10+25, 2020.

[17] S. Wu, *Research on Occlusion and Pose Robust Facial Landmark Localization*, Zhejiang University, Hangzhou, China, 2019.

[18] Y. H. Wu, W. Z. Cha, J. J. Ge et al., "Adaptive task sorting model based on improved linear regression method," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 48, no. 01, pp. 93–97+107, 2020.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[20] S. Park, S.-B. Lee, and J. Park, "Data augmentation method for improving the accuracy of human pose estimation with cropped images," *Pattern Recognition Letters*, vol. 136, 2020.

[21] Y. Z. Zhou, X. Y. Cha, and J. Lan, "Power system transient stability prediction based on data augmentation and deep residual network," *Journal of China Electric Power*, vol. 53, no. 01, pp. 22–31, 2020.

[22] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," *Computer Vision and Pattern Recognition*, vol. 9, no. 4, pp. 3476–3483, 2013.