Face Detection Based on Improved Multi-task Cascaded Convolutional Neural Networks

Siyu Jia, Ying Tian

Abstract—**With the development of deep learning and computer vision, face detection has achieved rapid progress owing. Face detection has several application domains, including identity authentication, security protection, media, and entertainment. Although multi-task cascaded convolutional neural networks (MTCNN) have high accuracy and robustness, the model has the disadvantages of large parameters and computational overhead in the real scene due to the complexity of the real scene and the constraints of hardware facilities. Therefore, the development of an improved network model is crucial. This paper improves the MTCNN model by reducing the number of parameters and the computational overhead and using better model parameters to locate the key points of the face. This model improves the accuracy and robustness of the face age estimation. The WiderFace and CelebA datasets are used for training. The final face detection accuracy reaches 98.7% while simultaneously reducing the number of model parameters to 70% under the same conditions. This model meets the application needs of modern society for face detection and demonstrates the efficiency and accuracy of the improved network model.**

Index Terms—**face detection, MTCNN, network optimization, object detection**

I. Introduction

THE face is the most important visual feature of the human body. It contains a wealth of personal information [1]. The human face has unique and clear advantages, such as convenience, safety, and non-contact. As a consequence, it has become an area of concern in biometric technology [2]. Furthermore, face detection is a crucial branch in the computer vision field. It is the first step in analyzing face information involving identity authentication, human-computer interaction, security monitoring, and social networking. [3]. With the widespread application domains of face detection and the constraints of the environment and hardware facilities, reducing the network model capacity and improving the detection effect are novel directions in the advancement in face detection.

Manuscript received May 18, 2023; revised November 27, 2023.

This work was supported by the Liaoning University of Science and Technology Student Innovation and Entrepreneurship Training Program project(Project No. X202310146068).

Siyu Jia is an undergraduate student majoring in software engineering at the College of Computer and Software Engineering, University of Science and Technology Liaoning, Liaoning 114051, China. (e-mail: 1044290021@qq.com)

Ying Tian is a professor of Software Engineering in the School of Computer and Software Engineering, University of Science and Technology Liaoning, Liaoning 114051, China. (Corresponding author to provide e-mail: astianying@126.com)

With the development of deep learning and extensive data networks, face detection based on deep convolutional neural networks (CNN) has been widely studied. The traditional face detection algorithms include AdaBoost [4], Haar-like [5], and HOG [6]. These algorithms use face geometric features and template matching methods for face detection but fail to balance detection speed and accuracy. In recent years, CNN have achieved higher accuracy. This developmental trend has brought unique computational costs to training and deployment. More precisely, due to the model size and computational cost, it is almost impossible to deploy the existing classic large models, such as AlexNet [7], VGG Net [8], and ResNet [9], on mobile phones, cars, and robots. Face detection can be considered a single task of target detection. Some target detection frameworks, such as R-CNN [10,11], SSD [12], YOLO [13], and FocalLoss [14], can be used for face detection with only slight improvement. The face detector based on CNN achieves higher performance than the traditional methods. It also provides a new benchmark for future methods. Although face detection based on deep learning has led to satisfactory results, as the development of deep CNN becomes more complex, the sharp increase in network parameters results in a computation expansion, significantly increasing the face detection processing time. As a consequence, some applications requiring high efficiency cannot be tackled.

II. Related Work

A. Depth Separable Convolution

The mobile lightweight network is proposed by Google's MobileNets [15], where deep separable convolution is developed for the first time. This model simultaneously considers real-time requirements and accuracy. The accuracy of the MobileNets model in the ImageNet classification task is 70.6% [16], while that of the VGG16 model is 71.5%. Although the accuracy of the VGG16 model is slightly higher than that of the MobileNets model [17], the number of calculations is 27 times higher, and the number of parameters is 32 times higher. In the traditional computer convolution calculation, due to the calculation characteristics of the convolution operation, the calculation of the corresponding area of all the channels will be considered in each step. This generates a large number of parameters and increases the computational load. In the depth separable convolution, the convolution calculation is first performed on the area of each channel. Then, the information exchange between the channels performs the complete separation of the intra-channel convolution and inter-channel convolution [18].

The depth separable convolution factorizes the original standard convolution operation into a deep convolution and a

full set point operation. In other words, it decomposes the original standard convolution into two convolutional layers. In contrast, each deep convolutional layer only convolves with each input channel using a convolution kernel. The point convolution layer is responsible for linearly combining the different channel feature map outputs by the previous layer, which can efficiently use the features of different channel feature maps at the same spatial position. This decomposition can significantly reduce the number of calculations and model parameters.

For the standard convolutional layer, the main factors affecting the number of calculations are the size and number of layers of the input network feature maps, the size and number of convolution kernels, the step size during the convolution operation, and whether the current feature maps or the matrix boundary is filled. Fig. 1 presents the standard convolution operation. Assuming that the input image is of $D_i \times D_i \times 3$ size, the feature maps of $D_0 \times D_0 \times n$ size are considered the output, where D_i and D_0 are the width and height of the input and output, respectively, and N is the number of channels of the output feature maps and the number of convolution kernels. The convolution kernel is of $D_k \times D_k \times 3 \times n$ size, where D_k represents its width and height. Without considering the influence of the bias term parameter, the computational cost of a standard convolutional layer is expressed as follows:

$$D_k \times D_k \times 3 \times N \times D_i \times D_i \tag{1}$$

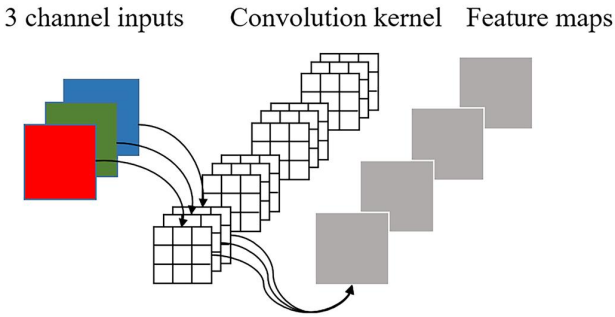

Fig. 1. Standard convolution operation

The depth separable convolution can be divided into deep convolution and point convolution. The left part of Fig. 2 presents the convolution operation of the deep convolution layer, where the convolution kernel is of $D_k \times D_k \times 3$ size. Here, D_k is its height and width, and 3 is the number of channels of the input image. The number of channels of each convolution kernel is 1. The number of convolution kernels equals the number of the input image channels. Each convolution kernel only performs convolution operations with a single channel of the input image. The following equation gives the computational cost of a deep convolutional layer:

$$D_k \times D_k \times 3 \times D_i \times D_i \tag{2}$$

The right part of Fig. 2 presents the convolution kernel structure of the point convolution layer, which is very similar to the standard convolution operation. The difference is that the convolution kernel size is 1×1×3, and 3 is the number of output channels of the previous layer. Consequently, the model output of the feature maps from the previous layer can be linearly combined, where N is the number of output channels. The computational cost of a point convolutional layer is computed as follows:

$$3 \times N \times D_i \times D_i \tag{3}$$

Under normal circumstances, a standard convolution is converted into a depth separable convolution, greatly reducing the calculation amount. The following equation gives the ratio of the calculation amount of the two methods:

$$\frac{D_k \times D_k \times 3 \times D_i \times D_i + 3 \times N \times D_i \times D_i}{D_k \times D_k \times 3 \times N \times D_i \times D_i} = \frac{1}{N} + \frac{1}{D_K^2} \tag{4}$$

B. MTCNN

The multi-task cascaded convolutional network (MTCNN) is a face detection framework based on cascaded CNN [19]. It uses multi-task learning to detect face frames and simultaneously locate five key facial points. As illustrated in Fig. 3, MTCNN contains three CNN. PNet is a small, full CNN [20]. The backbone consists of three convolutions and one pooling layer. It quickly screens out the face candidate regions from the multi-scale dense sliding box of the image pyramid and performs rough regression. RNet is a slightly larger network [20]. The backbone consists of three convolutions, two pools, and one full connection. It suppresses many false detection samples generated by PNet using finer classification and fine-tunes the face candidate boxes. ONet is the largest network [21]. The backbone consists of four convolutions, three pools, and one full connection. It uses a more robust model to perform the candidate boxes. It also eliminates a small number of wrong candidate samples and detects faces. The frame and five key facial points are accurately regressed. Among the three detections, non-maximum suppression is used to merge the candidate frames with more prominent intersections to avoid repeated detections. The MTCNN designs small, medium, and large CNNs in the dense, medium, and sparse detection stages. This helps to balance the computational overhead of each stage. It also uses a cascade structure that integrates many samples to be quickly eliminated and key samples to be gradually fine-tuned. This concept provides very effective balance and is currently the most efficient face detector.

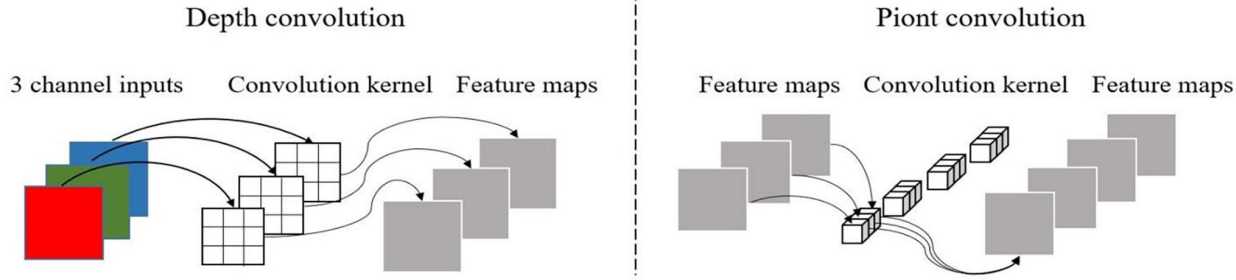

Fig. 2. Depth separable convolution operation

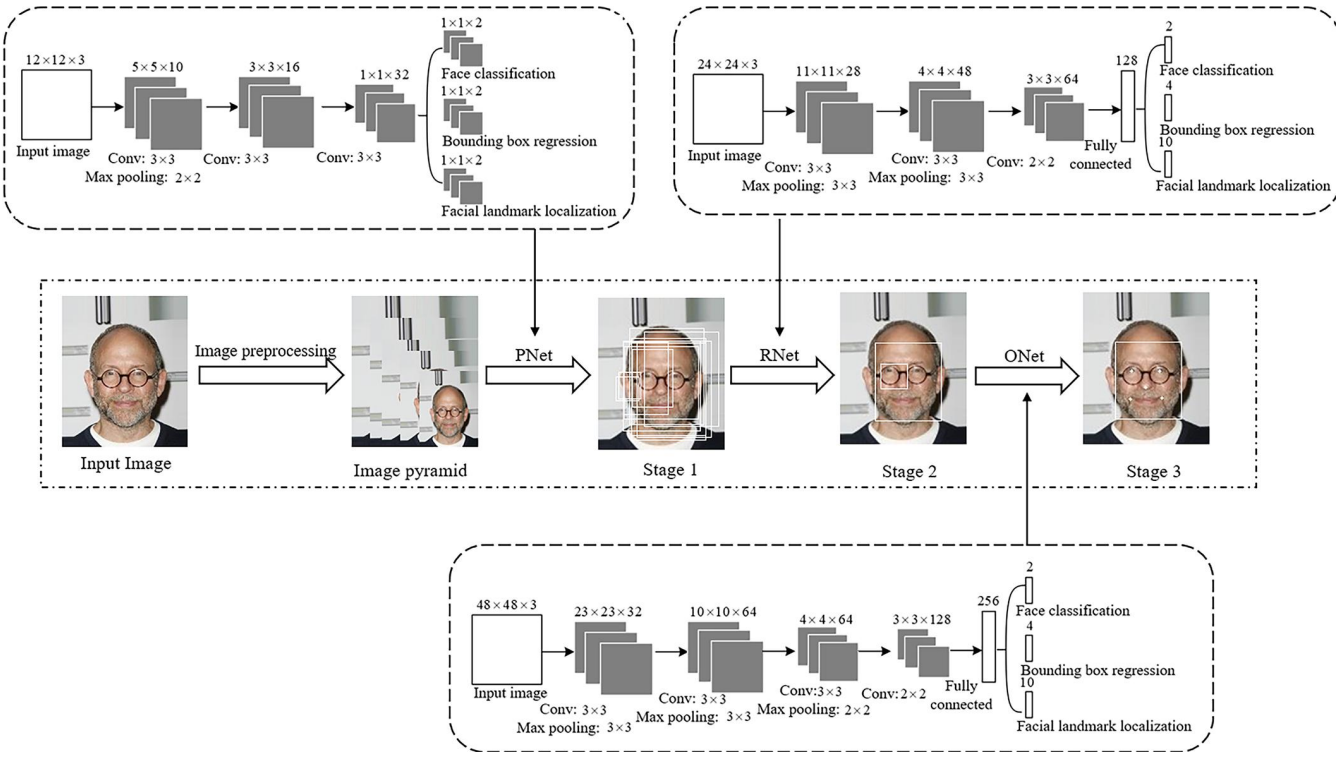

Fig. 3. MTCNN network structure

In a real-time face recognition and analysis system, it is necessary to compress face detection, facial critical point positioning, face alignment, face recognition, age, and gender recognition algorithms into a concise amount of time. Although MTCNN has a certain degree of efficiency, the detection time still occupies most of the time. Therefore, it is necessary to optimize the detection time of MTCNN. Based on the statistics of the time spent in each detection stage of MTCNN, the time spent in the PNet stage accounts for more than 60% of the detection time. Therefore, optimizing PNet is the key to accelerating MTCNN. Although the prediction cost of PNet for a single sample is less than that of RNet and ONet, when faced with the dense multi-scale sliding box in the image pyramid, too many prediction samples cause the overall speed of PNet to be low. Optimizing the detection speed of PNet is mainly based on image pyramid sparseness and a lightweight backbone network.

C. *Maxout Activation Function*

The Maxout activation function was proposed by Yann LeCun. in 2015 [22] and achieved excellent results in the ImageNet image classification task. The Maxout activation function is an activation function used in neural networks that can effectively improve the performance of the model. The core idea of the Maxout activation function is to divide an input vector into several subspaces, each of which is processed using the ReLU activation function, and then combine the outputs of these subspaces to obtain the final output. As shown in formula (5), the input vector is x, and the Maxout activation function will divide x into k different subspaces, each with a size of m. For each subspace, the Maxout activation function will calculate a weight matrix W and a bias vector b, and then use these two parameters to process the input vector using the ReLU activation function. Finally, the Maxout activation function will combine the outputs of all subspaces according to certain rules to obtain the final output.

$$h_i(\mathrm{x}) = \max_{\mathrm{j}\in[1,k]} z_{ij} \tag{5}$$

where $z_{ij} = \mathrm{x}^{T}{}_{\cdots ij} + \mathrm{b}_{ij}$,and $W \in \mathrm{R}^{\mathrm{d}\times m\times \mathrm{k}}$ and $b \in R^{\mathrm{m}\times \mathrm{k}}$.

Maxout activation function can be viewed as a learnable piecewise linear function, as shown in Figure 4. Assuming

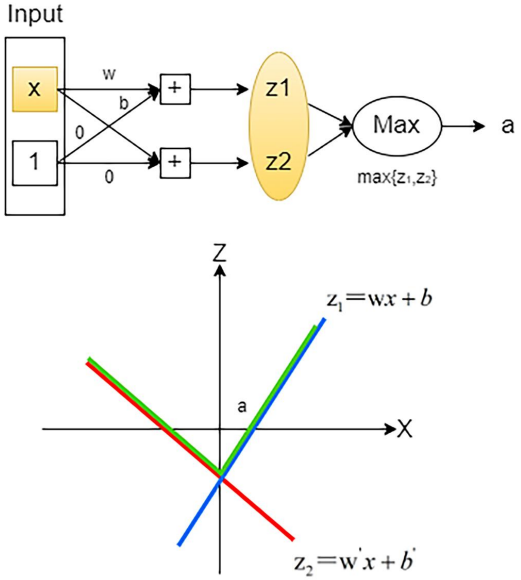

Fig. 4. Maxout activation function for a group of two neurons

there are two neurons in the output layer, the function with the highest value among z_1 and z_2 is selected to be passed to the activation function. Its fitting ability is strong. One of its advantages is that it can better express the complex patterns and structures in the input data because it shares weights among different neurons. In addition, Maxout

activation function can reduce the risk of overfitting and improve the model's generalization ability, which can be considered as the generalization of ReLU and Leaky ReLU activation functions. Compared with the ReLU activation function, Maxout activation function performs better.

III. Improved MTCNN

A. Figures and Tables

When MTCNN performs face detection, the input is an image, and the output is the bounding box where the face is located in the image. The MTCNN is based on CNN. It is usually only suitable for detecting faces within a specific size range. For example, PNet determines whether there are faces in a 12×12 size range. However, the face size in the input image is unknown. An image pyramid is first constructed to obtain images of different sizes [23]. Image scaling is then performed. The latter consists of scaling the face in the image to a suitable size that the network can detect. A face can be detected as long as its scale is almost 12×12. The reason for the slow speed of the PNet stage in MTCNN during dense sampling is that the image pyramid scale affects the process. PNet compromises on sampling density reduces the scale gap using dense image pyramids, and guarantees the recall rate of PNet in the detection stage. This paper adjusts the zoom factor at this stage. The scale factor is adjusted to $0.707 \approx \frac{\sqrt{2}}{2}$, so that the width and height become equal to the original value ($\frac{\sqrt{2}}{2}$), and the area becomes 1/2 of the original size.

B. Lightweight MTCNN

A series of improvements are performed to the MTCNN structure to make the MTCNN model more lightweight and better for face detection. PNet is a fully convolutional network. The ordinary convolution is first replaced with a deep separable convolution. This method can significantly reduce the number of network parameters and then replace the ReLU activation function with a better-performing Maxout activation function [24]. Finally, a BN layer is added [25], which allows the model to speed up the training and convergence speed of the network.

The introduction of deep separable convolution can greatly reduce the number of network parameters. However, the detection accuracy of the PNet model is reduced by 2%. The Maxout activation function is introduced to maintain the detection accuracy of the network model and improve the network performance. In addition, the BN layer avoids over-fitting and can also speed up the convergence of the loss function.

From this introduction to the MTCNN, it can be deduced that the network structure of RNet and ONet is similar. In terms of lightweight network improvement, a deep separable network is introduced. This consists of adding the dropout layer after the fully connected layer. The fully connected layer has the highest number of parameters. Adding the dropout layer after the fully connected layer can efficiently prevent overfitting. The lightweight structure of the three models is presented in Fig. 5.

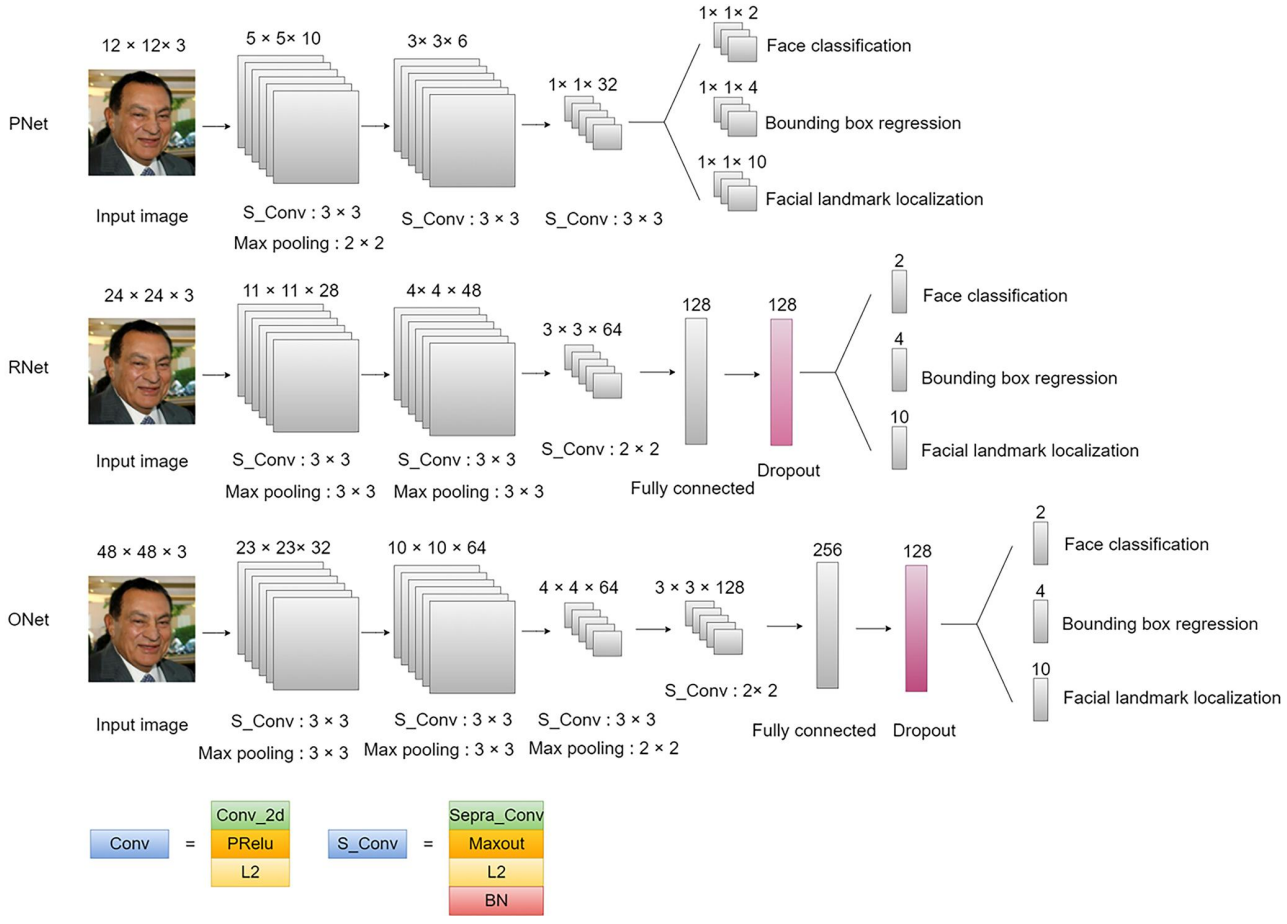

Fig. 5. Improved models of PNet、RNet and ONet

The convolutional layer changes of RNet and ONet are similar to those of PNet to reduce the number of parameters within the entire MTCNN model. The dropout layer is added after the fully connected layer to prevent the model from overfitting and improve the generalization ability of the model [23].

C. Core Algorithms

1) Loss function for face classification

Face detection mainly recognizes two types of facial images: those with faces and those without faces. The cross-entropy loss function is used to achieve this classification function. For each class of data, the cross-entropy loss function is computed as follows:

$$L_i^{\det} = -(y_i^{\det} \log(p_i) + (1 - y_i^{\det})(1 - \log(p_i))) \quad (6)$$

where p_i represents the probability that the picture to be classified is a face, and $y_i^{\det}$ denotes that the picture to be classified is a true and accurate label with a value range of $\{0,1\}$.

2) IoU

The Intersection over Union (IoU) is a crucial concept in the target detection algorithm. It defines the overlap between the prediction frame and the real frame. It is used to evaluate the positioning accuracy of the target detection algorithm. The cross-to-match ratio is computed as follows:

$$IoU = \frac{S_{Detection\,Result} \cap S_{GroundTruth}}{S_{Detection\,Result} \cup S_{GroundTruth}} \quad (7)$$

where $S_{Detection\,Result}$ is the network prediction frame and $S_{GroundTruth}$ is the artificially labeled real frame.

3) Bounding box regression loss function

The result is usually directly discarded when IoU is less than the set threshold. The method of the bounding box regression consists of predicting the offset between each prediction box and the real box, performing a series of fine-tuning prediction boxes, and making the prediction box closer to the real one. For the face image, the regression loss is calculated using the Euclidean distance:

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2 \quad (8)$$

where $\hat{y}_i^{box}$ is the coordinate value of the sample obtained by network prediction, and y_i^{box} is the coordinate value of the real and accurate sample x_i, namely the left, upper, right, and lower four points. Therefore, $y_i^{box} \in R^4$.

4) Regression loss function for landmark positioning

Similar to the boundary regression, the Euclidean distance between the predicted location of the landmark and the actual landmark is calculated, and the distance is minimized. For the face image x_i, the regression loss is calculated using the Euclidean distance:

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2 \quad (9)$$

where $\hat{y}_i^{landmark}$ is the landmark location obtained by network prediction, and $y_i^{landmark}$ represents the actual real landmark coordinates.

Since five points exist, each point has two coordinates. Therefore, $y_i^{landmark} \in R^{10}$.

5) NMS

The non-maximum suppression (NMS) aims to suppress the elements that are not maximum. In target detection, this method can quickly remove the boxes with low prediction scores and high overlap. In the initial stage of the model, that is, when using PNet and RNet, many overlapping candidate boxes will be framed. The non-maximum suppression algorithm will remove the candidate frames with relatively low prediction scores and high overlap; therefore, the prediction scores are relatively high. Finally, the tall candidate frame will be retained.

IV. Experiments

A. Experimental Dataset

The WiderFace dataset was released by the Chinese University of Hong Kong in 2016 [26]. It is the most authoritative dataset in the field of face detection. The WiderFace dataset contains 32,203 color pictures and 393,703 face annotation frames. The dataset fully covers the scale, posture, occlusion, expression, makeup, and lighting scenes, which are very difficult to detect, as shown in Fig.6. During the experiment, the dataset is divided according to the following standard: 50% for training, 10% for verification, and 40% for testing. According to the difficulty of detection, the WiderFace database divides the data into three levels: easy, medium, and difficult. During the evaluation, it will test the accuracy-recall curve and average accuracy of the detector on the three difficulty levels of the verification set and the test set.

Fig. 6. Samples of the WiderFace dataset

The CelebA dataset contains 202,599 face pictures of 10,177 celebrities [27]. It also contains personal facial feature point coordinates and 40 attribute markers. CelebA is openly provided by the Chinese University of Hong Kong. It is widely used for face-related computer vision training tasks and can also be used for face attribute identification training, face detection training, and landmark marking.

Fig. 7. Samples of the CelebA dataset

B. Experimental Results and Analysis

In the first stage of facial image processing, the original image will be zoomed in multiple times to obtain the image pyramid to make the face in the zoomed-in image close to the image scale (12×12) during the PNet training. According to the previously presented algorithm, the smaller the scale factor, the fewer the number of generated image pyramid layers and the smaller the amount of resize and PNet calculations.

As the scale factor decreases, the number of cycles in the first stage gradually decreases, the number of images of different scales processed in the image pyramid stage decreases, and the PNet processed images also decrease. However, if the scale factor is too small, many features will be lost. When the image pyramid is zoomed in, the width and height are changed to 1/2 of the original ones by default, and the area becomes 1/4 of the original one after being zoomed in. However, the zoom range of 1/4 is too large. Therefore, the area is scaled to 1/2 of the original size. Thus, the scale factor is adjusted to $0.707 \approx \frac{\sqrt{2}}{2}$, so that the width and height become equal to the original value ($\frac{\sqrt{2}}{2}$), and the area becomes 1/2 of the original size.

The model has the highest time percentage during the training phase. Furthermore, as TABLE I illustrates, the model parameters will be significantly lowered following improvement because PNet is a fully convolutional network.

TABLE I
APPROXIMATE AMOUNT OF CALCULATION OF PNET

Input size	Approximate amount of calculation of PNet performed once in total(1e6 FLOPs)	
	Original model	Improved model
300*300	165.60	45.87
480*640	565.25	148.52
1080*960	1907.71	458.25
2560*1920	9043.97	2120.33

Furthermore, there will be a significant reduction in the approximate computational load. As TABLE II demonstrates, many RNet and ONet parameters are taken from the fully connected layer. As a result, it is unclear how the modified model's parameters were reduced.

Figures 8 and 9 present the error and accuracy curves of each network layer function of the number of training rounds during the network training process. The falling curve represents loss and the rising curve represents accuracy. The abscissa denotes the number of training rounds, and the ordinate denotes the error and accuracy in the training process. The error is computed as follows:

$$total_{loss} = radio_{cls_{loss}} \times cls_{loss} + radio_{bbox_{loss}} \times bbox_{loss} + radio_{landmark_{loss}} \times landmark_{loss_{op}} + L2_{loss} \quad (10)$$

where $total_{loss}$ is the total error; cls_{loss} and $radio_{cls_{loss}}$ are the category loss and its corresponding weight, respectively; $bbox_{loss}$ and $radio_{bbox_{loss}}$ are the bounding box regression loss and its weight, respectively; $landmark_{loss_{op}}$ and $radio_{landmark_{loss}}$ are the key point regression loss and its weight, respectively; and $L2_{loss}$ represents the L2 regularization loss.

The improved MTCNN's total loss curve is shown in blue in Fig. 8, whereas the original MTCNN's total loss curve is shown in orange. Each MTCNN sub-neural network's convergence speed can be seen to be faster. The loss curve gradually flattens out as the number of training rounds rises, and the upgraded network effect performs less well than the original network.

The accuracy rate refers to the task's classification accuracy rate, namely the positive and negative classification accuracy rates. The average of the loss values and accuracy rates for each step within an epoch represents the loss value and accuracy rate for that particular epoch. The accuracy curves of the original MTCNN and the enhanced MTCNN are shown in Fig. 9. as the orange and blue curves, respectively. It is evident that the enhanced network outperforms the original one in terms of classification accuracy.

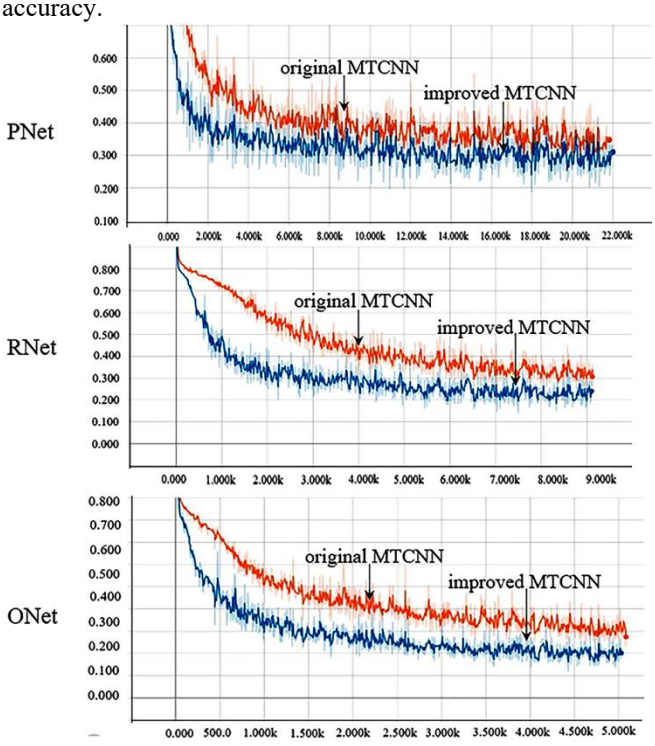

Fig. 8. The total_loss curve of each MTCNN layer

TABLE II
COMPARISON TABLE OF PARAMETER AND CALCULATION BEFORE AND AFTER IMPROVEMENT OF MTCNN

Network name	Total parameter quantity (thousands)		Approximate amount of calculation performed once in total(1e6 FLOPs)	
	Original model	Improved model	Original model	Improved model
Pnet	6.83	2.95	2920.63	693.24
Rnet	100.67	80.63	1.57	1.32
Onet	388.50	309.52	13.12	11.21

The accuracy rate is that of the classification task, that is, the classification accuracy rate of positive and negative. The loss value and accuracy rate corresponding to each epoch are the average of the loss values and accuracy rates of all the steps in the epoch. In Fig. 9, the orange curve represents the accuracy curve of the original MTCNN, and the blue curve denotes the accuracy curve of the improved MTCNN. It can be seen that the classification accuracy of the improved network is higher than that of the original one.

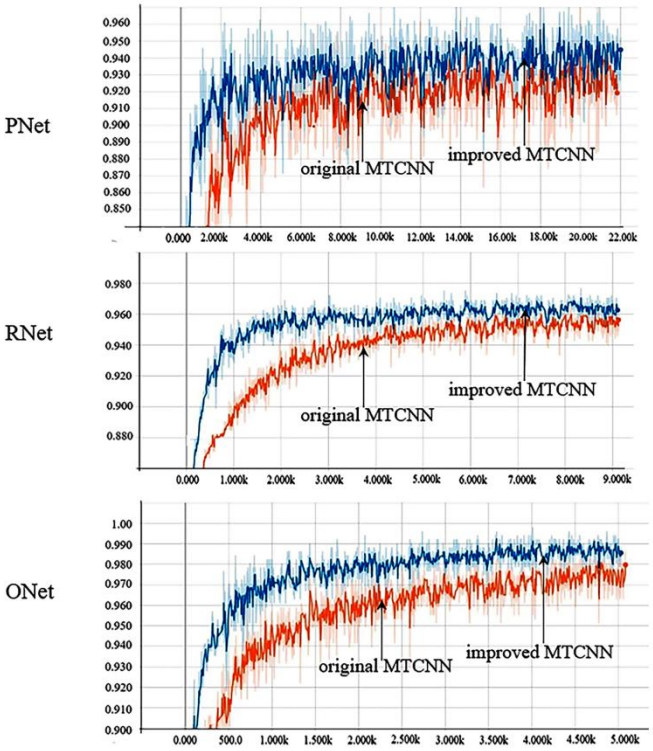

Fig. 9. Classification accuracy curves of each MTCNN layer

Table III presents the average loss rate and accuracy rate of MTCNN before and after weight reduction. The loss rate is the average value when the loss area is stable, while the accuracy rate is the average value when the accuracy rate tends to stabilize. It can be seen that the loss rate decreases to a certain extent, and the accuracy rate increases with the deepening of the network.

TABLE III
AVERAGE LOSS RATE AND ACCURACY RATE OF MTCNN, BEFORE AND AFTER LIGHTWEIGHT

	MTCNN			Lightweight MTCNN		
	PNet	Rnet	Onet	PNet	Rnet	Onet
Loss rate	0.35	0.3	0.28	0.31	0.27	0.25
accuracy rate	0.924	0.958	0.978	0.934	0.965	0.987

C. *Face Detection Experiment*

After the training is completed, the trained model is used to test the detection effect of the lightweight model. The obtained results are presented in Figs. 10 and 11. In the test diagram, the box represents the face frame detected by the model, the red dots in the box represent the five vital facial points of the detected face, and the numbers on the red box denote the detected face. It can be observed that the detection confidence of faces of varying sizes is different. Additionally, the detection confidence of the faces having different angles is varied.

Figure. 10 presents the face detection diagram of the WiderFace test set. Most of the WiderFace dataset images contain more than two human faces. Therefore, this dataset is mainly used for the face detection of multiple faces in an image. In detecting multiple faces, the confidence of the face detection with more obvious facial contours will reach the detection effect of 1. In contrast, the confidence in the detection of some faces will decrease. It can be deduced from the face detection results shown in the test set that the face detection effect of the lightweight MTCNN can meet most of the face detection requirements.

Fig. 10. Illustration of the face detection on the WiderFace test set

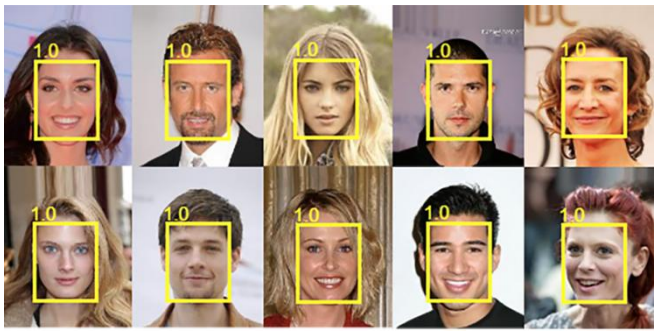

Fig. 11. Face detection diagram of the CelebA test set

Figure. 11 presents the face detection diagram of the CelebA test set. Because the face images of the CelebA dataset are all single-person images, most of the facial images are intact. In addition, a small number of cases have a slightly turned face. Therefore, accurate face detection can be performed using this dataset.

V. CONCLUSION

In this paper, the model is optimized using the MTCNN face detection concept after the images have been preprocessed. The suggested approach minimizes the number of model parameters, increases model performance, and lessens computational load. The experiments are carried out on the TensorFlow framework with Python. The WiderFace and Celeb A datasets are utilized for simulation and training verification. Under the same circumstances, the final accuracy rate of face detection can reach 98.7%, and the initial number of model parameters is decreased to 1.3 times. The experimental findings also show that the upgraded

MTCNN is more able to adapt to the demands of mobile devices and can more effectively meet face detection criteria. Finally, the suggested technique enhances face detection's effectiveness and efficiency, better satisfies face detection's application needs in contemporary culture, and has practical importance.

REFERENCES

[1] Yousaf A, Khan M. J, Siddiqui A. M, and Khurshid K,"A robust and efficient convolutional deep learning framework for age-invariant face recognition." *Expert Systems* vol. 37, no. 3, 2020, e12503.

[2] Chen Q, Yang L, Zhang D, and Huang S, "Face deduplication in video surveillance", *International Journal of Pattern Recognition and Artificial IntelligenceVol.* vol. 32, no. 3, 2018.

[3] Li H, Lin Z, Shen X, Brandt J, and Gang H, "A convolutional neural network cascade for face detection," *in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5325-5334.

[4] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," *in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. I-I.

[5] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *in Proceedings of the International Conference on Image Processing*, 2002, pp. I-I.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005),* vol. 2, 2005, pp. 886-893.

[7] Lin L, Zhang G, Wang J, Tian M, and Wu S, "Utilizing transfer learning of pre-trained AlexNet and relevance vector machine for regressionfor predicting healthy older adult's brain age from structural MRI", *Multimedia Tools and Applications*, 2021, pp. 24719–24735.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, vol. 2014.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4 inception-ResNet and the impact of residual connections on learning", *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1-7.

[10] Girshick Ross. "Fast r-cnn." *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440-1448.

[11] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 6, 2017, pp. 1137–1149.

[12] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, and Berg A C, "SSD: single shot multibox detector," *European Conference on Computer Vision. Springer, Cham*, 2016, pp. 21-37.

[13] R. Joseph and F. Ali, "Yolov3: an incremental improvement," *CoRR,* vol. 2018, 2018.

[14] Lin T Y, Goyal P, Girshick R, He K and Dollár P, "Focal 1oss for dense object detection", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 2017, no. 99, pp. 2999-3007.

[15] Howard, A.G, Zhu, M, Chen, B, Kalenichenko, D, Wang, W, Weyand, T, Andreetto, M, & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv, abs/1704.04861.

[16] Sherman T, Teng SW, Murshed M, Lu G, Sohel F and Paul M, "Enhanced Transfer Learning with ImageNet Trained Classification Layer ", *In Pacific-Rim Symposium on Image and Video Technology,* 2019, pp. 142-155, Springer, Cham.

[17] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR, abs/1409.1556.*

[18] Lu G, Zhang W, and Wang Z, "Optimizing Depthwise Separable Convolution Operations on GPUs," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 70-87, 1 Jan. 2022, doi: 10.1109/TPDS.2021.3084813.

[19] Zhang K, Zhang Z, Li Z, and Qiao Y, "Joint face detection and alignment using multitask cascaded convolutional networks", *IEEE Signal Processing Letters*, vol. 23, no. 10, 2016, pp. 1499-1503.

[20] Xie Y, Wang H, and Guo S, "Research on MTCNN Face Recognition System in Low Computing Power Scenarios", *Journal of Internet Technology*, vol. 21, no. 5, pp. 1463-1475, 2020.

[21] Chauhan A, Varghese B K, Rahman L A, Mohapatra V, and Badal T. "WIDER Face Challenge using Multi-Task Cascading Neural Network", In 2019 IEEE 9th *International Conference on Advanced Computing (IACC)*. December, 2019, pp. 188-192.

[22] Goodfellow, Ian J. et al. "Maxout Networks." International Conference on Machine Learning , 2013, pp. 1-8.

[23] Wu C and Zhang Y, "MTCNN and FACENET Based Access Control System for Face Detection and Recognition", *Aut. Control Comp. Sci*. 55, 2021, pp. 102–112.

[24] He K, Zhang X, Ren S, and Sun J, " Deep residual learning for image recognition", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] Liu Z, Li J, Shen Z, Huang G, Yan S, and Zhang C, "Learning efficient convolutional networks through network slimming", *In 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2755–2763. IEEE, 2017.

[26] Yang S, Luo P, Loy C C, and Tang X, "Wider face: A face detection benchmark", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5525-5533 2016.

[27] Sunhem W and Pasupa K, "An approach to face shape classification for hairstyle recommendation", *In 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 390–394. IEEE, 2016.