Parallel Multi-task Cascade Convolution Neural Network Optimization Algorithm for Real-time Dynamic Face Recognition

Bin Jiang[1], Qiang Ren[1], Fei Dai[2], Tian Zhou[2], and Guan Gui[2*]
[1]College of Electronic and Optical Engineering & College of Microelectronics, NJUPT, Nanjing, 210023 China
[2]College of Telecommunications and Information Engineering, NJUPT, Nanjing, 210003 China
*Corresponding authors: Guan Gui (e-mail: guiguan@njupt.edu.cn)

Received January 16, 2019; revised July 9, 2019; accepted January 14, 2020; published October 31, 2020

Abstract

Due to the angle of view, illumination and scene diversity, real-time dynamic face detection and recognition is no small difficulty in those unrestricted environments. In this study, we used the intrinsic correlation between detection and calibration, using a multi-task cascaded convolutional neural network(MTCNN) to improve the efficiency of face recognition, and the output of each core network is mapped in parallel to a compact Euclidean space, where distance represents the similarity of facial features, so that the target face can be identified as quickly as possible, without waiting for all network iteration calculations to complete the recognition results. And after the angle of the target face and the illumination change, the correlation between the recognition results can be well obtained. In the actual application scenario, we use a multi-camera real-time monitoring system to perform face matching and recognition using successive frames acquired from different angles. The effectiveness of the method was verified by several real-time monitoring experiments, and good results were obtained.

Keywords: multi-task cascaded convolutional neural networks, real-time dynamic face recognition, facial feature algorithms

http://doi.org/10.3837/tiis.2020.10.011 **ISSN : 1976-7277**

1. Introduction

Both face detection and alignment are indispensable parts of many face-based applications, such as facial recognition [1] [2] [3] [36] [43] and facial analysis [4] [5]. Dynamic recognition poses a great challenge because of the visual changes in the face plus the effects of posture and light. And in many embedded applications or IoT applications, the timeliness of face recognition is also an important indicator. We hope to accomplish this task with the least time and resource costs.

In Section 2, we list some previous work on face detection and recognition. Section 3 discusses the optimization algorithm and model architecture of parallel multi-task convolutional neural networks in real-time dynamic multi-camera applications. We describe a multitasking cascade convolutional neural network with FaceNet method to improve the efficiency of face recognition and map the output of each core network in parallel into a compact Euclidean space, where the distance represents the similarity of facial features, which can be quickly Identify the target face. In Section 4, the performance of the parallel multi-task convolutional neural network was tested using FDDB and CASIA-FaceV5 datasets. The algorithms in different experiments were compared and evaluated. Section 5 summarizes the work done and suggests further directions for improvement.

2. Related Work

Nowadays, convolutional neural network (CNN) [6] [7] [8] has been applied to achieve good results for the applications such as image classification [9] [41] and face recognition.

Viola and Jones [10] proposed a cascade face detector which took advantage of Haar-like features and AdaBoost method to train cascaded classifiers and achieve good performance with real-time efficiency. Y. Sun, X. Wang and X. Tang [11] implements the face annotation algorithm based on CNN, and finds that the optimization of deep convolution network depends heavily on the structure and initial weight. H. Li, Z. Lin, X. Shen, *et al*. [12] proposes a convolutional neural network based cascade structure and introduces a CNN-based calibration phase after each detection phase of the cascade. It has strong recognition capabilities while maintaining high performance. D. Chen, S. Ren, Y. Wei, *et al*. [13] proposed a new face detection method. The core idea is to combine face alignment and detection, and to observe the aligned face shape to provide better features for face classification. Liu *et al*. [14] proposed a two-stream transformer network (TSTN) approach for face detection technology based on dual video stream that decomposes the video stream into a time stream and a spatial stream. X. Yu *et al*. [15] solves the problem of facial landmark location from the aspect of a single camera. A two-level cascade deformable model is proposed. It is a group sparse learning method to induce the selection of the most significant facial landmarks. By introducing a 3D facial shape model and using Procrustes analysis, the pose-free facial landmark initialization is realized. E. Zhou *et al*. [16] proposes a method of locating a wide range of face markers using a coarse-to-fine convolutional network cascade in which each network layer is trained to locally refine the face generated by the previous network layer. A subset of the logo. In addition, each level predicts explicit geometric constraints to correct current network level inputs. B. Shi *et al*. [17] presented a deep regression face alignment method that estimates the initial face shape from the entire image, while the subsequent layers iteratively update the shape. Combined with standard derivatives and numerical approximations, all layers are able to backproper error differentiation, so that standard backpropagation can be applied to jointly learn parameters from all layers.

However, quite a few studies indicated that this detector may degrade significantly in real-world applications with larger visual variations in human faces even if more advanced features and classifiers are included. Most of the previously developed face recognition methods often ignore the effect of face angle and lighting. In order to address these problems, several algorithms have been proposed but they do not fully address the effects of face angle and lighting. In addition, there is a lack in the performance improvement of current training sample classifier. It is necessary to develop a classification algorithm that can extract facial features with higher precision and achieve better performance in matching facial features. Let it connect the output of each core CNN in parallel to perform real-time target matching on the detected object.

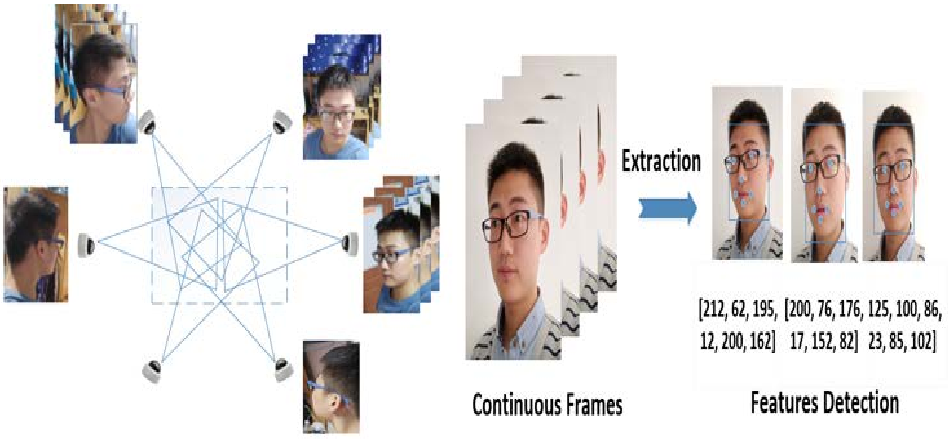

Fig. 1. Continuous images are acquired in real time by camera in real scene, and the feature matrix of key frame face images is calculated by MTCNN algorithm. The source of the photograph is the photograph taken by the author himself.

In this paper, we propose a novel cascading-parallel architecture to lift the dynamic face recognition performance. The algorithm consists of three steps. In the first step, multiple cameras are used and successive frames are obtained from different angles; the candidate form is retained using a multi-task cascaded convolutional neural network (MTCNN) [18] [19] and the face in the video stream [20] [21] is extracted. In the second step, the FaceNet [22] [23] extraction is used. The 128 feature values of the face are stored in an array. In the third step, the face datasets are classified and matched using a k-nearest neighbor (KNN) method [24] [33] to achieve dynamic face recognition.

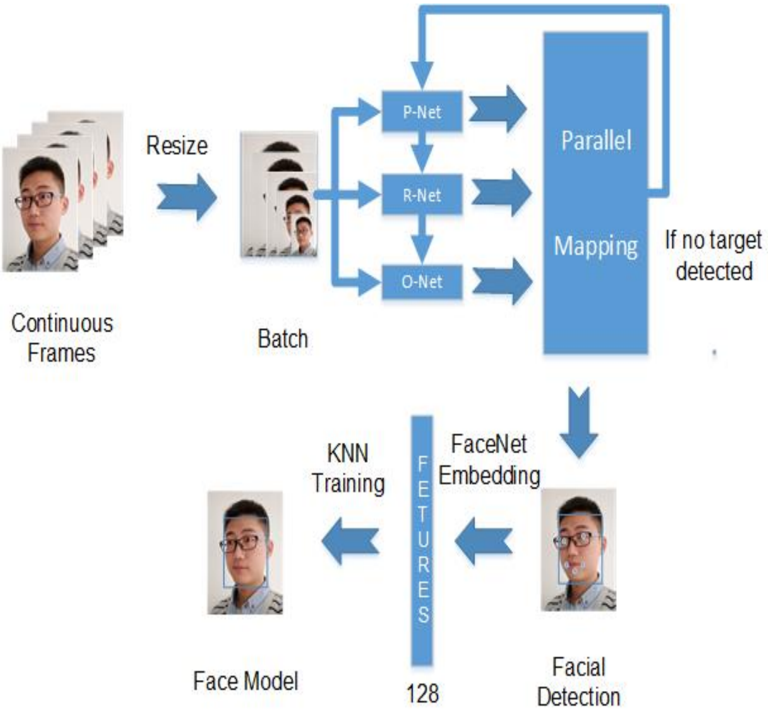

Fig. 2. Framework of the real-time dynamic face recognition.

3. Algorithm Principle and Model Formula

3.1 Multi-task cascaded convolutional neural networks

The MTCNN first uses a full CNN, proposal net (P-Net) to calculate the candidate forms and boundary regression vectors. At the same time, the candidate form is using the bounding box to calibrate. Then, we use the network management system (NMS) [29] method to remove overlapping windows. Subsequently, the picture containing the candidate form determined by the P-Net is trained in the refine net (R-Net) network; the network uses the full connection method for training. We use the bounding box vector to finetune the candidate form and use the NMS to remove the overlapping form. Finally, the network structure is more convolutional than the R-Net. The function is the same as that of the R-Net. It only shows the key position of the face while removing the overlapping candidate window.

The MTCNN feature descriptor includes three parts, face/non-face face classifier, bounding box regression, and landmark location.

$$\mathrm{L}_i^{det} = -\left(y_i^{det}\log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(p_i))\right) \tag{1}$$

where $y_i^{det} \in \{0,1\}$. Eq. (1) is a cross-entropy loss function for face classification, where p_i is the probability of the face and y_i^{det} is the real tag of the background.

$$\mathrm{L}_i^{box} = \left\|\hat{\mathbf{y}}_i^{box} - \mathbf{y}_i^{box}\right\|_2^2 \tag{2}$$

where $\mathbf{y}_i^{box} \in \mathbb{R}^4$. Eq. (2) is the regression loss calculated by the Euclidean distance. In the equation, $\hat{\mathbf{y}}$ is predicted through the network, $\mathbf{y}$ represents the actual real background coordinates and the quad (upper left $\mathbf{x}$, upper left $\mathbf{y}$, long, wide).

$$\mathrm{L}_i^{landmark} = \left\|\hat{\mathbf{y}}_i^{landmark} - \mathbf{y}_i^{landmark}\right\|_2^2 \tag{3}$$

where $\mathbf{y}_i^{landmark} \in \mathbb{R}^{10}$. As with the boundary regression, the Euclidean distance between the predicted landmark position and the actual real landmark is calculated and the distance is minimized. $\hat{\mathbf{y}}$ is predicted through the network and $\mathbf{y}$ is the actual real landmark coordinate. Since there are a total of 5 points and 2 coordinates for each point, $\mathbf{y}$ is a ten-tuple. The multiple input source training is defined as follows:

$$\min \sum_{i=1}^{N} \sum_{j\in\{det,box,landmark\}} \alpha_j \beta_i^j L_i^j \tag{4}$$

where $\beta_i^j \in \{0,1\}$. The objective of the training and learning process is to minimize Eq. (4), where N is the number of training images, α_j indicates the importance of the task, β_i^j is the sample tag, and L_i^j is the above loss function Eq. (4).

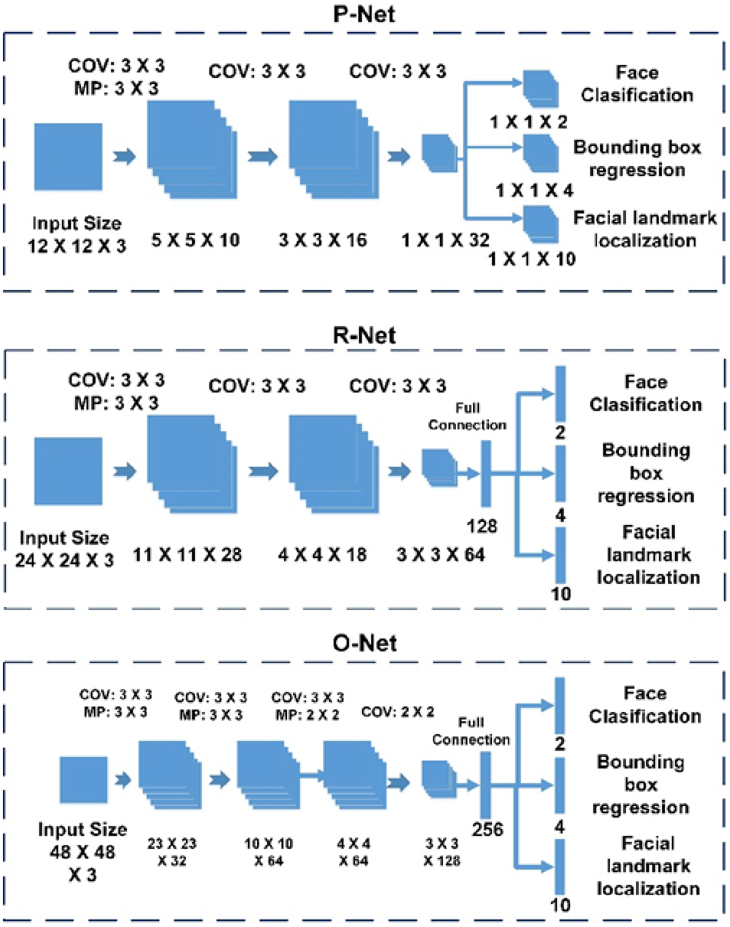

Fig. 3. The proposed MTCNN-based facial feature detection method is divided into three steps: 1) proposal net, 2) refine net, and 3) output net.

A comparison of the computing speed and verification accuracy of CNNs at each core level shown in **Table 1**.

Table 1. Comparison of CNNs speed and verification accuracy.

Group	CNN	400 x Forward Propagation	Validation Accuracy
Group 1	P-Net	0.038s	94.70%
Group 2	R-Net	0.491s	95.10%
Group 3	O-Net	1.478s	95.10%

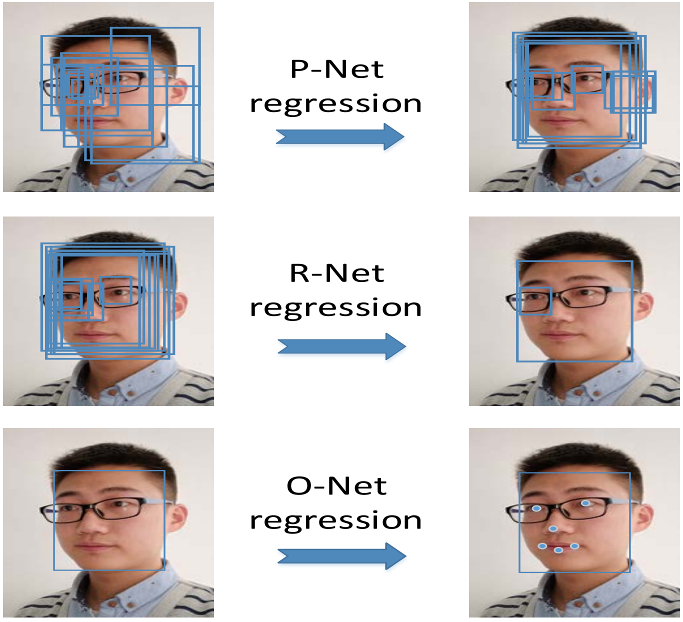

Fig. 4. Flowchart of the proposed three-step method.

In order to improve the algorithm results, in the process of training, only the gradient of the first 70% of the samples is transmitted backward at a time to ensure that the transmitted numbers are valid. This approach is similar to a latent support vector machine (SVM) [35] but it embodies the end-to-end learning of deep learning.

3.2 Analysis of the time complexity of the MTCNN

The time complexity determines the training/prediction time of the model. If the complexity is too high, model training and prediction consume too much time and ideas cannot be verified quickly, the model cannot be improved, and the prediction is slow. Therefore, the amount of time required by the algorithm and the number of basic operations performed are at most one constant factor. We analyze the time complexity of each convolution layer cascade after the MTCNN algorithm model is logically analyzed.

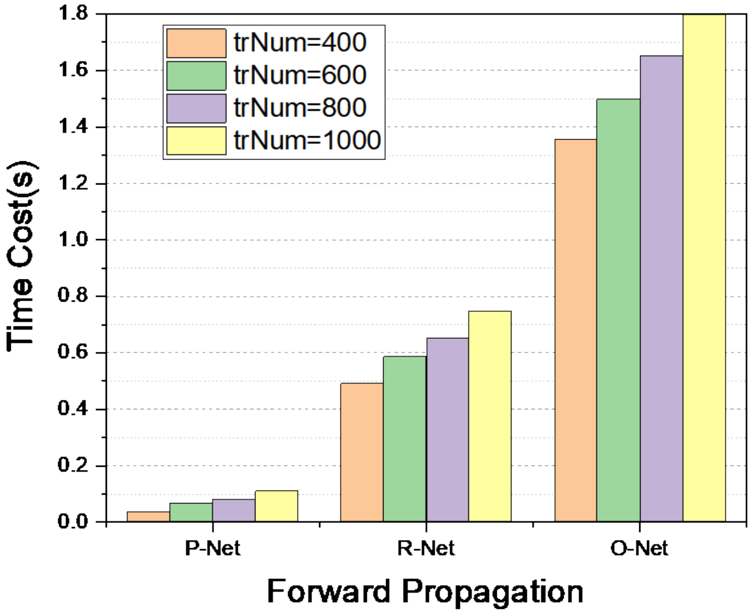

Fig. 5. Time cost of each section of the MTCNN.

First, the MTCNN uses a convolution kernel for the convolution operation of the input image. *M* represents the output characteristics of the convolution kernel. *K* represents the size of each convolution kernel. If C_{in} represents the number of input channels and C_{out} represents the number of output channels, then the time complexity is determined by the area of the output feature map $\mathbf{M}^2$, the area of the kernel $\mathbf{K}^2$, and the number of input and output channels.

The size of the output feature map is determined by the size of the input matrix **X**, the size of the convolution kernel **K**, **Padding** and **Stride**; therefore, the function is expressed as:

$$\mathbf{M} = \frac{\mathbf{X-K+2*Padding}}{\mathbf{Stride}} + 1 \tag{5}$$

To simplify the number of variables in the expression, it is assumed that the input image and convolution kernel are square; therefore, the time complexity can be expressed as:

$$\text{Time} \sim \text{O}(\ \mathbf{M}^2 * \ \mathbf{K}^2 * \ C_{in} \ * \ C_{out}\ ) \tag{6}$$

Assuming that the neural network has a convolution layer *D*, l represents the l-th convolution layer, *C* stands for the number of kernels, so C_{in} is the number of output channels in the $(l-1)$-th convolution layer. Therefore, the time complexity of the cascaded CNN is the accumulation of the time complexity of all coiling layers:

$$\text{Time} \sim \text{O}(\ \sum_{l=1}^{\text{D}} \mathbf{M}_l^2 * \ \mathbf{K}_l^2 * \ C_{l-1} * \ C_l) \tag{7}$$

Algorithm 1. Algorithm of cascade training and joint face alignment

```
Input: training images {$\boldsymbol{x}_i$}, image labels {$\boldsymbol{y}_i$}, basic prior shapes $\hat{\boldsymbol{S}}_i$ for positive images , $\boldsymbol{y}_i$ = 1
Output: weak learning samples {C $R_k^t$ }, taxonomy threshold {$\theta_k^t$}.
Initialize the face shapes $S_i^0$ in the images to a random disturbance in the  the window of $\boldsymbol{x}_i$
Initialize face classification parameters $f_i$ = 0
for t = 1 to T do
    for k = 1 to K do
        for each image i do
            calculate its weight $w_i$
        end for
        select a ramdom point for regression calculation
        compute  a structure of regression tree C$R_k^t$
        for each leaf do
            try to set classification parameters
        end for
        for each image i do
            renew its regression parameters as $f_i = f_i + CR_k^t(\boldsymbol{x}_i, S_i^{t-1})$
        end for
        use {$f_i$} to set the bias $\theta_k^t$
        remove images whose $f_i$ < $\theta_k^t$ from training samples list
    end for
    compute the face shape increments of regression tree leaves
    compute $S_i^t$ for all of images
end for
```

3.3 An improved parallel multi-task cascade convolutional neural network model

Under the existing MTCNN framework, the face detection tasks for different illuminations and angle conditions have been greatly improved. However, because of the limited number of output channels, the face recognition performance is not particularly good. Therefore, we integrate the original algorithm into the network architecture of parallel mapping FaceNet to effectively increase the output of the facial feature map and greatly improve the accuracy of the face recognition.

FaceNet performs a mapping from a facial image to a compact Euclidean space, where the distance corresponds directly to the measure of facial similarity. The spatial distance is directly related to the similarity of the pictures. Different images of the same person have a small distance and images of different people have a large distance. As long as the mapping is determined, the relevant face recognition task becomes simple. FaceNet directly uses the loss function of large margin nearest neighbor (LMNN) [34] based on triplets to train the neural network.

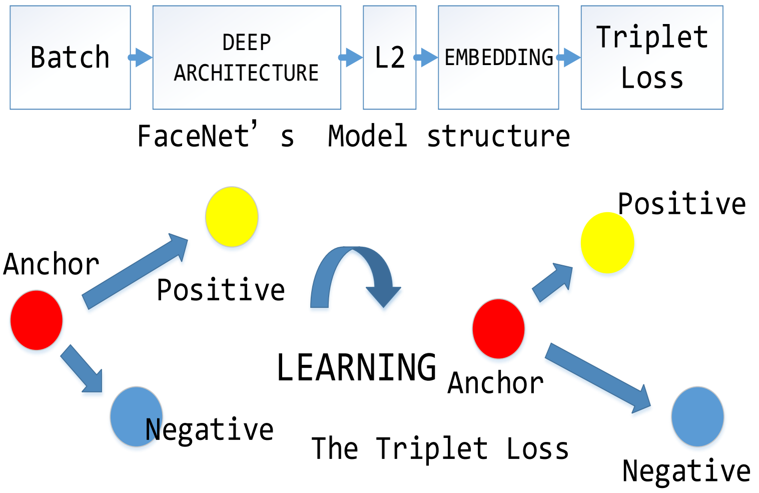

Fig. 6. FaceNet network architecture is composed of input layer and multi-layer convolution neural network, then normalized by L2 [30] function, and finally formed face embedded. As shown here is the three loss diagram of FaceNet network training process.

The network directly outputs a 128-dimensional vector space.

$$\left\|\mathbf{x}_i^a - \mathbf{x}_i^p\right\|_2^2 + \boldsymbol{\alpha} < \|\mathbf{x}_i^a - \mathbf{x}_i^n\|_2^2, \forall\left(\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{x}_i^n\right) \in \mathcal{T} \tag{8}$$

The loss that is being minimized is then

$$\sum_i^N \left[\left\|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^p)\right\|_2^2 - \|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^n)\|_2^2 + \boldsymbol{\alpha}\right]_+ \tag{9}$$

where $\boldsymbol{\alpha}$ is the positive/negative boundary. The choice of triplets is very important for the convergence of the model. For $\mathbf{x}_i^a$, we need to select different pictures of the same individual $\mathbf{x}_i^p$ to calculate argmax $\mathbf{x}_i^p \left\|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^p)\right\|_2^2$. At the same time, we also need to select images of different individuals $\mathbf{x}_i^n$ to calculate argmin $\mathbf{x}_i^n \left\|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^p)\right\|_2^2$. During practical training, it is unrealistic to calculate argmin and argmax across all training samples and training convergence is difficult due to incorrectly labeled images. Therefore, we filter by calculating the subset's argmin and argmax every n steps.

Next we will describe the parallel architecture of the whole system, that is, the neural network composed of MTCNN and parallel FaceNet.

First, we set up a fully connected network at the input level, which is a neural network layer with two output layers. The input is averagely pooled to 12×12 vector. After a series of convolution operations, the final convolution layer will output a vector of 1×16×1×1. Each feature point output from the two output layers is expressed in the form of probability distribution, that is, $\boldsymbol{p} = (\boldsymbol{p_0}, \boldsymbol{p_1})$ represents face and non-face, $\boldsymbol{t}^k = (\boldsymbol{t}_x^k, \boldsymbol{t}_y^k, \boldsymbol{t}_w^k, \boldsymbol{t}_h^k)$ for the possibility of predicting each face.

Then, we use the multi-task convolutional neural network to calculate the loss of the upper layer of the fully connected network as the input of three different loss functions, and use the bounding box regression method to jointly optimize the branches. Among that, the softmax loss is used for the classification and the L2 paradigm is used to regularize the bounding-box regression to achieve smoothness:

$$\mathrm{L}(\boldsymbol{p},\boldsymbol{u},\boldsymbol{t}^{u},\boldsymbol{v}) = \mathrm{L}_{cls}(\boldsymbol{p},\boldsymbol{u}) + \lambda[\boldsymbol{u} \geq 1]\mathrm{L}_{loc}(\boldsymbol{t}^{u},\boldsymbol{v}) \tag{10}$$

where $\mathrm{L}_{cls}(\boldsymbol{p},\boldsymbol{u}) = -\,log\,\boldsymbol{p}_u$ is a log loss for the class u.

We set λ = 0.1, because after reading papers and a lot of experiments, we find that this parameter will be applicable to most convolutional neural networks. And four coordinates for the regression offsets that are shown as following:

$$\boldsymbol{t}_x^* = (\boldsymbol{x}^* - \boldsymbol{x}_p)/\boldsymbol{w}_p \tag{11}$$

$$\boldsymbol{t}_y^* = (\boldsymbol{y}^* - \boldsymbol{y}_p)/\boldsymbol{h}_p \tag{12}$$

$$\boldsymbol{t}_w^* = log\,(\boldsymbol{w}^*/\boldsymbol{w}_p) \tag{13}$$

$$\boldsymbol{t}_h^* = log\,(\boldsymbol{h}^*/\boldsymbol{h}_p) \tag{14}$$

where x and y are the two coordinates representing the central area of the box and w and h represents the width and height of the box respectively. The proposed box and ground truth box are represented by variables $\boldsymbol{x_p}$, and $\boldsymbol{x}^*$. So we can optimize many uncertain face recognition targets obtained by learning, and then obtain the target frame reduction as shown in **Fig. 4** to achieve better recognition results.

Through continuous training, the regression deviation will be normalized into unit variance. So that the efficiency of the algorithm can be greatly improved. This optimization algorithm is also applicable to other loss functions in the system.

Then we will associate the output of the vectors $\mathbf{12} \times \mathbf{12}$ above with the output of the vectors $\mathbf{24} \times \mathbf{24}$ of the other branch in a fully connected form. That is, the output of $\mathbf{24} \times \mathbf{24}$ is a 128-dimensional vector. We can name it 24 *fc*, and then 24 *fc* is fully connected to the vectors of $\mathbf{16} \times \mathbf{16} \times \mathbf{1} \times \mathbf{1}$ above. The connection of this layer is named 12-24 *fc*.

Here we will set 0.1 as a classification threshold, which is used to better extract the areas we want from the upper layer network that meet the target characteristics. This threshold is well classified in the previous single cascade branch, and this threshold is also applicable to output the loss of classification and bounding regression.Similar to the algorithms described in the preceding two paragraphs, the final branch takes a 48x48 vector as input, outputs a 256-dimensional vector, and performs a fully connected operation with 12-24 *fc*. As in 12-24 *fc*, we also set a classification threshold. This parameter will determine the final output recognition result. Similar to the above test, we find that 0.003 is the most applicable.

By combining weight functions with these loss functions, we can get a new cascade loss function:

$$\boldsymbol{L_{joint}} = \boldsymbol{\lambda_1 L_{x12}} + \boldsymbol{\lambda_2 L_{x24}} + \boldsymbol{\lambda_3 L_{x48}} \tag{15}$$

where $\boldsymbol{L_{x12}}$, $\boldsymbol{L_{x24}}$ and $\boldsymbol{L_{x48}}$ represent the different losses of branches, which is calculated by Eq. (1). $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, and $\boldsymbol{\lambda_3}$ are the loss weights of each branch above.

The last step is the joint training of MTCNN and parallel FaceNet. We deeply analyzed the limitations of the training process of MTCNN algorithm, designing an architecture for joint operation of MTCNN and FaceNet. This architecture is shown in **Fig. 2**. When each branch of MTCNN gives an output, it will be used as parallel mapping input to the FaceNet network. The FaceNet network convolutes all inputs into a 128-dimensional vector output, which is the characteristic points of 128 target regions. These 128-dimensional vectors are then processed by KNN neural network to assist in target recognition, so as to achieve more accurate recognition of the target face.

4. Analysis of Results

We used multiple cameras in the interior to recognize pedestrians from different angles. The camera captures the pedestrian photos in real time, detects and corrects faces using the MTCNN model, and then uses these corrected faces as inputs to the trained FaceNet model; these inputs become the feature matrix with 128 eigenvalues. Finally, KNN training is performed on these feature matrices for face recognition. To train the data set of the KNN, we are using CASIA-FaceV5's 4,000 Asian face photos, which also includes 1000 face photos of our team members. In the training process, the Intersection-over-Union (IoU) ratio of $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ is 0.3 non-human face, 0.65 face, 0.4 face part, and 0.3 landmark; the training sample ratio is negative sample: positive sample: part sample: landmark = 3:1:1:2. We continue to appear in front of the camera, using the variable n to indicate the number of people identified and the variable m to indicate the number of times the team members were accurately identified. We also used several other algorithms to compare the accuracy of the face recognition from different angles.

4.1 CASIA-FaceV5 dataset for face recognition

In this section, We first use the CASIA-FaceV5 [32] face dataset training set to test our algorithm architecture. All the pictures in this dataset are captured by Logitech USB camera. Volunteers who provide face data include workers, attendants and researchers. The face changes of all the pictures in the dataset include different lighting, posture, facial expressions and glasses. We assign these images according to the data set allocation ratio mentioned above, that is, the training sample ratio is negative sample: positive sample: part sample: landmark = 3:1:1:2.

In order to obtain a relatively accurate and stable recognition rate, we use several typical face recognition algorithms in the data set for many tests, and use the average accuracy and the time cost of training to evaluate the performance of each algorithm on the CASIA-FaceV5 training set. The test results are shown in the following **Table 2**.

Table 2. The results of the test using the CASIA-FACEV5 dataset.

Algorithm	**Accuracy(%)**	**Training Time (s)**	**Time per batch (s)**
Joint fasterRCNN [27]	93.4±0.2	0.55	1.375×10^{-3}
Faceness [25]	92.1±0.2	1.58	3.95×10^{-3}
Video-based face alignment [26]	88.7±0.2	1.65	4.125×10^{-3}
CCF [28]	87.4±0.2	0.35	0.875×10^{-3}
MTCNN-FaceNet (Ours)	**94.20±0.4**	**0.67**	1.675×10^{-3}

The efficiency and accuracy of each algorithm tested are shown in the **Table 2**. The test results show that our MTCNN-FaceNet algorithm is nearly 2.5 times faster than the video-based Face Alignment and Faceness algorithms. And the accuracy is improved by approximately 10 and 5 percentage points, respectively. Obviously, our MTCNN-FaceNet has better performance and efficiency, and has more significant architectural improvements than other mainstream face recognition schemes.

The test results of the comparison between MTCNN-FaceNet with different lamda and the performances of MTCNN-FaceNet are shown in the following figures. **Fig. 8** shows the accuracy and **Fig. 9** shows the time required for recognition**.** The experimental results in **Fig. 8** and **Fig. 9** show that the time cost of *lamda* = 0.2 is more than 50% faster than that of *lamda* = 0.05 for the training time of the samples and the decrease in the average accuracy is about 1.5%.

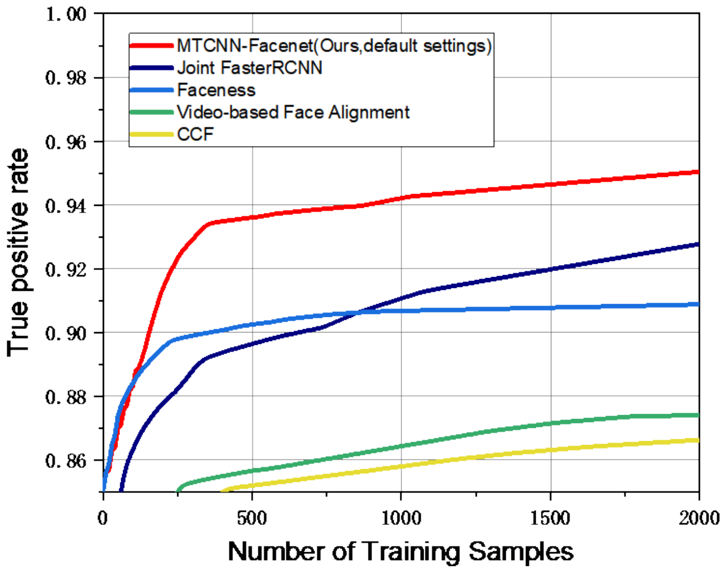

Fig. 7. Evaluation on CASIA-FACEV5.

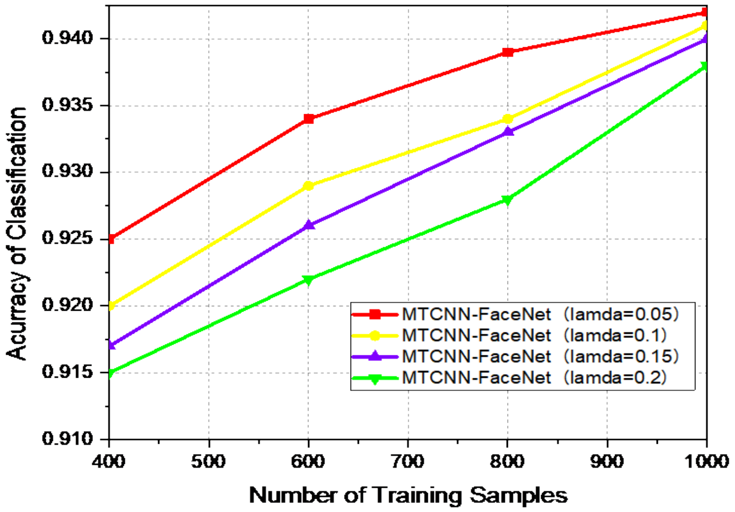

Fig. 8. Accuracy of the face recognition using the CASIA-FaceV5 dataset.

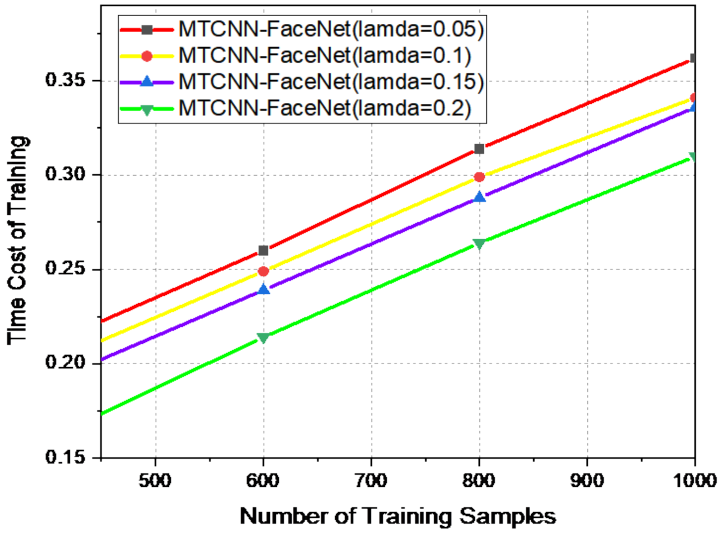

Fig. 9. The time required for recognition using the CASIA-FaceV5 dataset.

4.2 FDDB face dataset for face recognition

The face detection dataset FDDB [31] contains 2,845 images, which includes 5171 faces with different occlusions, different poses, different environmental light and different blurred faces. This allows for calibration of the face area with ellipses and grayscale and color maps.

Similar to the data processing method of the CASIA-FaceV5 training set, we also use several other typical face recognition algorithms to test, and use the average accuracy and the average time cost of training samples to evaluate their performance.

Table 3. The results of the test using the FDDB dataset.

Algorithm	**Accuracy(%)**	**Training Time (s)**	**Time per batch (s)**
Joint fasterRCNN [27]	86.8±0.02	0.65	2.167×10^{-3}
Faceness [25]	86.7±0.02	1.28	4.27×10^{-3}
Video-based face alignment [26]	86.6±0.02	1.05	3.5×10^{-3}
CCF [28]	86.4±0.02	0.51	1.7×10^{-3}
MTCNN-FaceNet (Ours)	**88.2±0.02**	**0.45**	1.5×10^{-3}

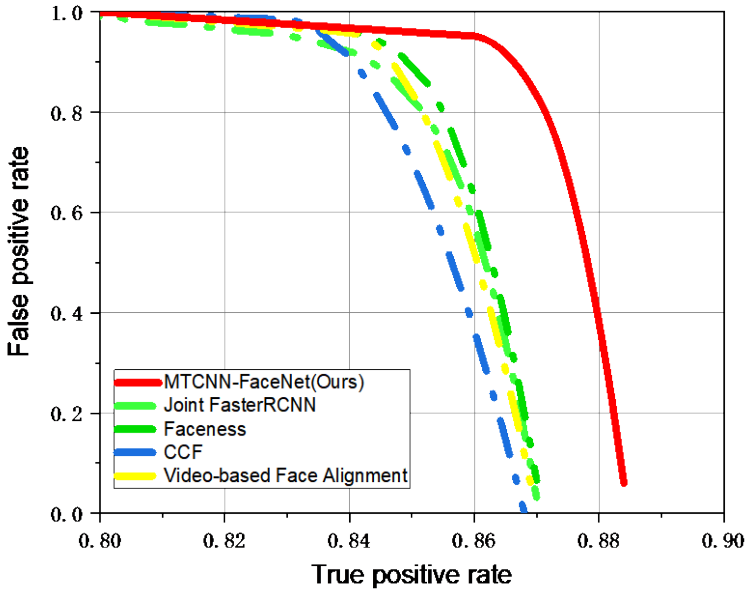

Fig. 10. Evaluation on FDDB dataset.

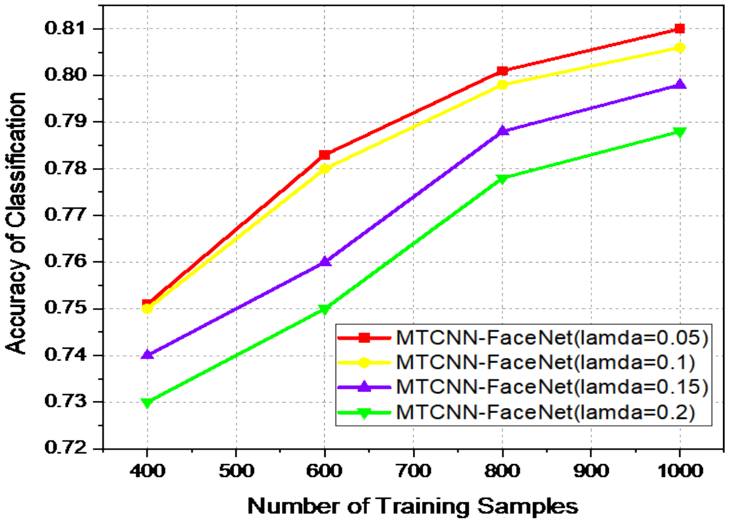

Fig. 11. Accuracies of the face recognition using the FDDB dataset.

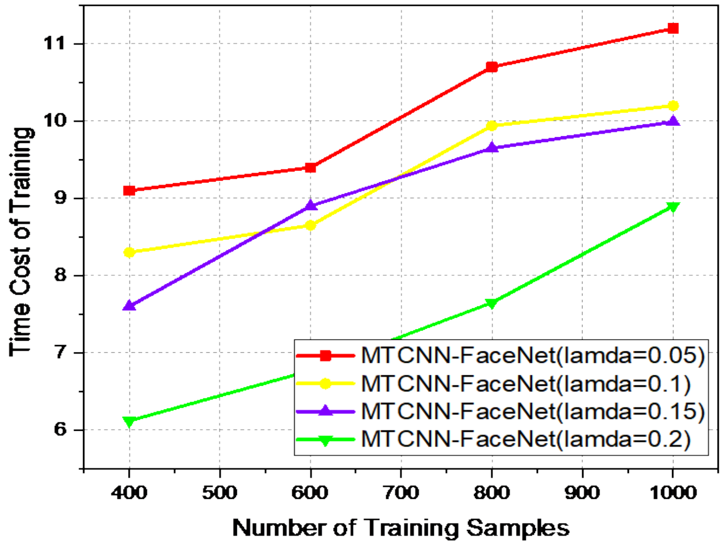

Fig. 12. The time required for recognition using the FDDB dataset.

Table 3 shows that our method has the best performance in the experiments using the FDDB face database. In terms of accuracy, our MTCNN-FaceNet architecture has the highest accuracy. As shown in the chart, the accuracy of our algorithm is 1.5 percent higher than that of Faceness algorithm, 1.6 percent higher than that of Video-based Face Alignment algorithm, and 1.4 percent higher than that of the Joint FasterRCNN algorithm. In terms of time cost, the MTCNN-FaceNet is much faster than the Video-based Face Alignment and Faceness by approximately 2.3 times and 2.84 times, respectively. The experimental results clearly show that the MTCNN-FaceNet architecture has better performance and is more robust because it has a good recognition effect on different data sets.

The experimental results shown in **Fig. 11** and **Fig. 12** indicate that the time cost of *lamda* = 0.2 is more than 50% faster than that of *lamda* = 0.05 for the training time of the samples and the decrease in the average accuracy is about 2%. However, due to the black box property of neural network, some unknown random values will appear in the process of operation, which makes the code running result unstable, such as the results in **Fig. 12** indicate an unstable performance. In spite of this, the performance is good regarding the tradeoff between accuracy and time cost.

5. Conclusion

In many current embedded terminals or IoT applications, the efficiency of real-time dynamic face recognition is greatly affected by the angle and illumination of the video stream. In addition, it is difficult to improve the performance of the training sample classifier, and it takes a long time to iteratively calculate the target face matching result. Therefore, we propose a method of parallel calculation of MTCNN-FaceNet, which uses the inherent correlation between detection and calibration to enhance facial recognition performance, improve the real-time performance of the whole system through parallel architecture, and shorten the time to recognize face targets. At the same time, the real-time parallel feedback can be used to compare the timely changes of the two frames before and after, so as to reduce the influence of the face angle and illumination on the recognition result in the video stream. In the future, we plan to use feature selection for further experiments and include more detailed areas for further research, including an automated selection (choose the correct *k*) algorithm to optimize the number of clusters by further refining MTCNN and k-means, In particular, the combination of methods and the accuracy of clustering are improved by further adjusting the hyperparameters.

References

[1] Y. Ren, Z. Wang, and M. Xu, "Learning-based saliency detection of face images," *IEEE Access*, vol. 5, pp. 6502–6514, 2017. Article (CrossRef Link)

[2] G. Ghinea, R. Kannan, and S. Kannaiyan, "Gradient-orientation-based PCA subspace for novel face recognition," *IEEE Access*, vol. 2, pp. 914–920, 2014. Article (CrossRef Link)

[3] M. Mei, J. Huang, and W. Xiong, "A discriminant subspace learning based face recognition method," *IEEE Access*, vol. 6, pp. 13050–13056, 2017. Article (CrossRef Link)

[4] M. Z. Uddin, M. M. Hassan, A. Almogren, A. Alamri, M. Alrubaian, and G. Fortino, "Facial Expression Recognition Utilizing Local Direction-Based Robust Features and Deep Belief Network," *IEEE Access*, vol. 5, pp. 4525–4536, 2017. Article (CrossRef Link)

[5] Y. Ding, Q. Zhao, B. Li, and X. Yuan, "Facial expression recognition from image sequence based on LBP and Taylor expansion," *IEEE Access*, vol. 5, pp. 19409–19419, 2017. Article (CrossRef Link)

[6] M. Matsugu, K. Mori, M. Ishii, and Y. Mitarai, "Convolutional spiking neural network model for robust face detection," in *Proc. of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, vol. 2, pp. 660–664, 2002. Article (CrossRef Link)

[7] X. Zhang, M. Peng, and T. Chen, "Face recognition from near-infrared images with convolutional neural network," in *Proc. of International Conference on Wireless Communications & Signal Processing*, pp. 1–5, 2016. Article (CrossRef Link)

[8] H. Wu, K. Zhang, and G. Tian, "Simultaneous face detection and pose estimation using convolutional neural network cascade," *IEEE Access*, vol. 6, pp. 49563–49575, 2018. Article (CrossRef Link)

[9] W. Zhao *et al.*, "Superpixel-based multiple local CNN for panchromatic and multispectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4141–4156, 2017. Article (CrossRef Link)

[10] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004. Article (CrossRef Link)

[11] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3476–3483, 2013. Article (CrossRef Link)

[12] G. Li, Haoxiang and Lin, Zhe and Shen, Xiaohui and Brandt, Jonathan and Hua, "A convolutional neural network cascade for face detection," in *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5325-5334, 2015. Article (CrossRef Link)

[13] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," *Lect. Notes Comput. Sci.*, vol. 8694 LNCS, no. PART 6, pp. 109–122, 2014. Article (CrossRef Link)

[14] H. Liu, J. Lu, J. Feng, and J. Zhou, "Two-stream transformer networks for video-based face alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2546–2554, 2018. Article (CrossRef Link)

[15] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Proc. of 2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 1944–1951, 2013. Article (CrossRef Link)

[16] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. of IEEE International Conference on Computer Vision Workshops*, pp. 386–391, 2013. Article (CrossRef Link)

[17] B. Shi, X. Bai, W. Liu, and J. Wang, “Face alignment with deep regression,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 1, pp. 183–194, 2018. Article (CrossRef Link)

[18] Yizhang Xia, Bailing Zhang, and F. Coenen, “Face occlusion detection based on multi-task convolution neural network,” in *Proc. of 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 375–379, 2015. Article (CrossRef Link)

[19] X. Li *et al.*, “DeepSaliency: Multi-task deep neural network model for salient object detection,” *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, 2016. Article (CrossRef Link)

[20] H. A. Elsalamony, “Automatic video stream indexing and retrieving based on face detection using wavelet transformation,” in *Proc. of 2010 2nd International Conference on Signal Processing Systems*, vol. 1, pp. V1-153-V1-157, 2010. Article (CrossRef Link)

[21] S. Ding, Y. Li, J. Zhu, Y. F. Zheng, and D. Xuan, “Sequential sample consensus: A robust algorithm for video-based face recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 10, pp. 1586–1598, 2015. Article (CrossRef Link)

[22] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 07-12-June 2015. Article (CrossRef Link)

[23] K. Heath and L. Guibas, “Facenet: Tracking people and acquiring canonical face images in a wireless camera sensor network,” in *Proc. of 2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 117–124, 2007. Article (CrossRef Link)

[24] M. A. Abuzneid and A. Mahmood, “Enhanced human face recognition using LBPH descriptor, multi-KNN, and back-propagation neural network,” *IEEE Access*, vol. 6, pp. 20641–20651, 2018. Article (CrossRef Link)

[25] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Faceness-Net: Face detection through deep facial part responses,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1845–1859, 2018. Article (CrossRef Link)

[26] Kuang-Chih Lee, J. Ho, Ming-Hsuan Yang, and D. Kriegman, “Video-based face recognition using probabilistic appearance manifolds,” in *Proc. of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I-313-I–320, 2003. Article (CrossRef Link)

[27] B. Liu, W. Zhao, and Q. Sun, “Study of object detection based on Faster R-CNN,” in *Proc. of 2017 Chinese Automation Congress (CAC)*, pp. 6233–6236, 2017. Article (CrossRef Link)

[28] W. Sibanda and P. Pretorius, “Comparative study of the application of central composite face-centred (CCF) and Box–Behnken designs (BBD) to study the effect of demographic characteristics on HIV risk in South Africa,” *Netw. Model. Anal. Heal. Informatics Bioinforma.*, vol. 2, no. 3, pp. 137–146, Sep. 2013. Article (CrossRef Link)

[29] J. Ma, L. Zhang, S. Zhang, and X. Yao, “Vulnerability analysis of the optical network NMS,” *International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 1185–1187, 2012. Article (CrossRef Link)

[30] C. Su and C. Tseng, “L1/L2 difference in phonological sensitivity and information planning — Evidence from F0 patterns,” in *Proc. of 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 1–5, 2016. Article (CrossRef Link)

[31] V. Jain, E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," *UMass Amherst Technical Report*, 2010. Article (CrossRef Link)

[32] S. Z. Li, D. Yi, Z. Lei, and S. Liao, “The CASIA NIR-VIS 2.0 face database,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 348–353, 2013. Article (CrossRef Link)

[33] Q. Liu and C. Liu, “A novel locally linear KNN model for visual recognition,” in *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1329–1337, 2015. Article (CrossRef Link)

[34] F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang, “A kernel classification framework for metric learning,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 9, pp. 1950–1962, Sep. 2015. Article (CrossRef Link)

[35] S. Pang, D. Kim, and S. Y. Bang, “Face membership authentication using SVM classification tree generated by membership-based LLE data partition,” *IEEE Trans. Neural Networks*, vol. 16, no. 2, pp. 436–446, 2005. Article (CrossRef Link)

[36] T. Zhou, S. Yang, L. Wang, J. Yao, and G. Gui, “Improved cross-label suppression dictionary learning for face recognition,” *IEEE Access*, vol. 6, pp. 48716–48725, 2018. Article (CrossRef Link)

[37] Z. Jiang, Z. Lin, and L. S. Davis, “Label consistent K-SVD: Learning a discriminative dictionary for recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013. Article (CrossRef Link)

[38] D. Wang and S. Kong, “A classification-oriented dictionary learning model: Explicitly learning the particularity and commonality across categories,” *Pattern Recognit.*, vol. 47, no. 2, pp. 885–898, Feb. 2014. Article (CrossRef Link)

[39] X. Wang and Y. Gu, “Cross-label suppression: A discriminative and fast dictionary learning with group regularization,” *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3859–3873, Aug. 2017. Article (CrossRef Link)

[40] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, “Decorrelation of neutral vector variables: Theory and applications,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 1, pp. 129–143, Jan. 2018. Article (CrossRef Link)

[41] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, “Variational Bayesian matrix factorization for bounded support data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, Apr. 2015. Article (CrossRef Link)

[42] Z. Ma *et al.*, “IEEE Access special section editorial: Recent advantages of computer vision,” *IEEE Access*, vol. 6, pp. 31481–31485, 2018.

[43] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016. Article (CrossRef Link)

[44] Y. Goh, Z. Ho, C. Ng, and Y. Goh, "Enhanced CNN-based plant growing-stage classification using additional information carried in an additional channel," *IEIE Transactions on Smart Processing & Computing*, vol. 8, no. 3, pp.171-177, June, 2019. Article (CrossRef Link)

Bin Jiang is pursuing B.Sc. degree in Electronic Engineering from Nanjing University of Posts and Telecommunications, Nanjing, China. His current research is Deep Learning based Image Processing.

Qiang Ren is pursuing B.Sc. degree in Electronic Engineering from Nanjing University of Posts and Telecommunications, Nanjing, China. His current research is Deep Learning based Image Processing.

Fei Dai is pursuing B.Sc. degree in Electronic Engineering from Nanjing University of Posts and Telecommunications, Nanjing, China. His current research is Deep Learning based Image Processing.

Tian Zhou is currently pursuing his Master's degree of communication engineering at Nanjing University of Posts and Telecommunications, Nanjing China, from 2018. His research interest is dictionary learning, deep learning and convex optimization.

Guan Gui received the Dr. Eng degree in Information and Communication Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012. From 2009 to 2014, he joined the wireless signal processing and network laboratory (Prof. Adachi laboratory), Department of Communications Engineering, Graduate School of Engineering, Tohoku University as for research assistant as well as postdoctoral research fellow, respectively. From 2014 to 2015, he was an Assistant Professor in Department of Electronics and Information System, Akita Prefectural University. Since 2015, he has been a professor with Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China.

He is currently engaged in research of deep learning, compressive sensing and advanced wireless techniques. Dr. Gui has published more than 200 international peer-reviewed journal/conference papers and received night best paper awards, e.g., ICNC 2018, ICC 2017, ICC 2014 and VTC 2014-Spring. He received Member and Global Activities Contributions Award (2018), and Top Editor Award of IEEE Transactions on Vehicular Technology (2020). He was also selected as for Jiangsu Specially-Appointed Professor (2016), Jiangsu High-level Innovation and Entrepreneurial Talent (2016), Jiangsu Six Top Talent (2018), Nanjing Youth Award (2018). Dr. Gui was an Editor of Security and Communication Networks (2012~2016). He has been the Editor of IEEE Transactions on Vehicular Technology, since 2017, the Editor of IEEE Access, since 2018, the Editor of Physical Communication, since 2019, the Editor of KSII Transactions on Internet and Information Systems since 2017, the Editor of Journal of Communications, since 2019, and the Editor-in-Chief of EAI Transactions on Artificial Intelligence, since 2018. He is IEEE Senior Member.