Journal of Computers Vol. 34 No. 1, February 2023, pp. 145-155
doi: 10.53106/199115992023023401011

Non-uniform Illumination Face Recognition Method Based on Improved MobileNetV1

Tian-Tian Chen[1,2], Guo-Jun Lin[1,2*], Hong-Jie Zhang[1,2], Hong-Rong Jing[1,2], Shun-Yong Zhou[1,2]

[1] Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China
{1450935593, 386988463} @qq.com
[2] School of Automation and Information Engineering, Sichuan University of Science and Engineering, Zigong 643000, China
{2369095868, 2366464126} @qq.com

Received 19 April 2022; Revised 17 July 2022; Accepted 11 August 2022

Abstract. Under ideal illumination conditions, the existing face recognition algorithms can obtain better recognition results. However, under non-uniform illumination conditions, the performance of the face recognition algorithm will be greatly reduced. Moreover, the common facial image classification models have low precision and speed. To address these problems, this paper proposes a non-uniform illumination face recognition method based on the improved MobileNetV1, and constructs a lightweight feature extraction network with MobileNetV1 as the core. To reduce the influence of non-uniform illumination on the face image, we use the MSRCR algorithm to preprocess the face image. A lightweight I-MobileNet model is proposed. We introduce the attention module into the MobileNetV1 model, which enhances the feature extraction capability of the network and improves the recognition performance. At the same time, the model parameters are optimized with Arcface loss function to increase the distance between classes and reduce the distance within classes. Compared with the original MobileNetV1 model, our method shows a significant improvement in recognition accuracy. In comparison with other network models, experiments conducted on the Extended Yale B dataset with the I-MobileNet model demonstrate the effectiveness of the method.

Keywords: face recognition, non-uniform illumination, MobileNetV1, attention module

1 Introduction

In practice, under non-uniform lighting conditions, the captured face images can suffer from shadows and uneven brightness, which can cause interference with face recognition. Studies have shown that lighting can seriously affect the performance of face recognition systems. For example, the difference in face images of the same person under different lighting conditions is even greater than the difference between the different people [1]. The solution to this problem has two forms: on the one hand, it starts at the source of image acquisition, using methods such as laser scanning or infrared imaging. However, this solution requires professional image acquisition equipment. On the other hand, the performance of the face recognition system can be improved by a series of face images processing, which has been affected by non-uniform illumination. At present, most face recognition systems use ordinary cameras to collect images. Therefore, many face recognition algorithms have emerged at home and abroad to cope with changes in illumination. These algorithms can be grouped into three categories: face illumination modelling methods, illumination enhancement methods and invariant feature methods [2]. Face illumination modelling methods represent the variation in illumination between face images in a suitable low-dimensional subspace, and then estimate the model parameters from the face characteristics. Although this type of method can reduce the effect of illumination on faces, the modelling effort is large, time-consuming and the algorithm is complex. Illumination enhancement methods are used to first restore the face image to a consistent illumination state in the case of uneven illumination, and then perform face recognition. However, all such methods are too simple and difficult to resolve well in practice due to the interference caused by complex light. The invariant feature method is to extract features that are insensitive to illumination changes from the face image to represent the face, reducing the impact of uneven illumination on the accuracy of the face recognition algorithm. However, these features are not robust enough for drastic light changes, and the face image under uniform illumination is not obtained from these features, which makes it impossible to manually verify the recognition results of the algorithm in practice.

* Corresponding Author

In recent years, deep learning has become increasingly popular. It has been widely used in computer vision tasks, such as object detection, image recognition, and image segmentation. It has also achieved considerable success in the field of face recognition. In 2014, Facebook proposed the DeepFace [3] model and obtained a recognition rate of 97.35% on the LFW face recognition dataset. The algorithm uses 3D faces for face alignment and then feeds the aligned images into an 8-layer convolutional neural network to obtain facial features. Google proposed the FaceNet model in 2015 [4], which achieved a recognition rate of 99.63% on the LFW dataset. The algorithm maps face images to Euclidean space with the help of a convolutional neural network, allowing face similarity to be associated with a measure of spatial distance. Although the above algorithms have a high recognition rate on the LFW face recognition dataset. In real scenarios, the face images collected by the camera often have a variety of non-uniform illumination interference, and the above algorithms cannot recognize the face accurately. In response to this situation, N. P. Ramaiah [5] and Y. H. Kim et al. [6] first used deep convolutional neural networks to extract illumination robust features of non-uniformly illuminated face images. Then use common recognition algorithms for subsequent face recognition. But the overall effect is not ideal. T. Sun et al. [7] proposed an end-to-end deep learning method, which is more natural for photos with uniform original illumination. However, this method cannot solve the problem of shadows and highlights on the original image.

1.1 Purposes

Currently, the recognition rate of face images under non-uniform illumination needs to be improved. The existing network models have slow recognition speed and insufficient feature extraction. In order to address these issues, this paper uses a lightweight convolutional neural network - MobileNetV1. The traditional network model has a good deal of parameters and low precision. And it is not suitable for use in embedded devices. The MobileNetV1 network has low parameter number and computational cost, and runs fast. To improve the recognition rate of face images under non-uniform illumination, this paper improves the MobileNetV1 network. The attention mechanism is embedded into the network, enabling the model to concentrate on necessary features and enhancing the feature extraction capability of the model. The ReLU activation function is used in the depth-separable convolutional layer of the MobileNetV1 network. To retain more feature information in the image, a linear output is used instead of the original ReLU activation function. In addition, the original Softmax classifier is replaced with Arcface classifier, which increases the spacing between classes and reduces the intra-class spacing.

Our contributions can be summarized as follows: 1) A lightweight I-MobileNet model is proposed to obtain better results on the Extended Yale B dataset. 2) The attention module is embedded in the network to make the model focus on important features and enhance the feature extraction ability of the network. 3) The model parameters are optimized with the Arcface loss function for face recognition.

The content of this paper is structured as follows. The first section is the introduction. The second section is illumination preprocessing. This section includes two parts, the first part is the introduction of the Retinex theory, the second part is the introduction of the MSRCR algorithm, using the algorithm for illumination preprocessing. The third section is the network model. This section includes three parts: the first part is the network structure of MobilenetV1, the second part introduces the attention module, and the third part describes the Arcface loss function. The fourth section is the I-MobileNet network structure, the fifth section is the experiment. The sixth section is the conclusion, and the last section is the acknowledgement.

2 Image Illumination Preprocessing

2.1 The Retinex Theory

In 1963, E. H. Land proposed the Retinex [8] algorithm, which is a common method for enhancing images based on scientific experiments and scientific analysis. Removing or weakening the incident component on the original image and retaining only the reflected properties of the object in the image is the main idea of the Retinex algorithm. Its mathematical form is shown in Eq. (1).

$$I(x,y) = R(x,y) \cdot L(x,y). \tag{1}$$

Where $I(x,y)$ represents the original image, $L(x,y)$ is the illumination component, and $R(x,y)$ represents the reflected component, determined by the object's properties, and is the light-invariant component. To be able to

separate $R(x,y)$ and $L(x,y)$, the original image needs to be transformed into the logarithmic domain for processing, as shown in Eq. (2).

$$I''(x,y) = \ln I(x,y) = \ln R(x,y) + \ln L(x,y) = R''(x,y) + L''(x,y). \tag{2}$$

Here, $I''(x,y)$ is the image after the original image has been transformed into the logarithmic domain, and $R''(x,y)$ and $L''(x,y)$ are the light invariant component and light component on the logarithmic domain, respectively. From Eq. (2), since $L(x, y)$ corresponds to the low frequency part of the image, $L''(x,y)$ can be obtained by passing $I''(x,y)$ through a low-pass filter. Then, subtract $I''(x,y)$ and $L''(x,y)$ to get $R''(x,y)$. Finally, inverse transform $R''(x, y)$ to get $R(x,y)$.

The general Retinex theory approximates $L(x,y)$ by Gaussian filtering of the original image. $R(x,y)$ can be expressed as Eq. (3).

$$\ln R(x,y) = \ln I(x,y) - \ln[l(x,y) * G(x,y)]. \tag{3}$$

Here, $G(x,y)$ is the Gaussian kernel, and $R(x\ y)$ can be restored at this point. Eq. (3) is the SSR algorithm in the Retinex theory.

The MSR algorithm is a classical algorithm developed for image enhancement based on the Retinex theory. It can achieve better visual effects by adding multiple scales. The expression is:

$$R_{MSR}(x,y) = \sum_{k=1}^{K} \varphi_k \{\ln I(x,y) - \ln\left[I(x,y) * G(x,y)\right]\}. \tag{4}$$

Where K is the total number of scales, generally taken as 3, and φ_k is the scale factor and satisfies $\sum_{k=1}^{K} \varphi_k = 1$.

2.2 MSRCR Algorithm

The MSRCR [9] algorithm introduces a color recovery factor into the MSR algorithm. The color of the image after enhancement is generally better than that of the MSR algorithm. Its expression is:

$$R_m(x,y) = C_m(x,y) \cdot R_{MSRm}(x,y). \tag{5}$$

Here, $C_m(x,y)$ is the colour recovery factor for the m'th color channel, which is generally obtained using the R, G and B channel values of the original image and can be expressed as:

$$C_m(x,y) = \omega\left(\frac{f_m(x,y)}{\sum_{n\in R,G,B} f_n(x,y)}\right). \tag{6}$$

Where $f_m(x,y)$ represents the m'th channel of the image, ω denotes the mapping function, and its parameters are the relative ratios of the R, G, and B components in the original image.

This paper uses the MSRCR algorithm to process face images under non-uniform illumination and compares it with other algorithms. The following are the image processing results of the three algorithms, SSR, MSR and MSRCR. Comparing the enhancement effects of the three Retinex algorithms, it can be seen in Fig. 1 that the MSRCR algorithm clearly improves the quality of the image, enhancing the detail and contrast, while SSR and MSR are not as good as the MSRCR algorithm in terms of detail in dark areas.

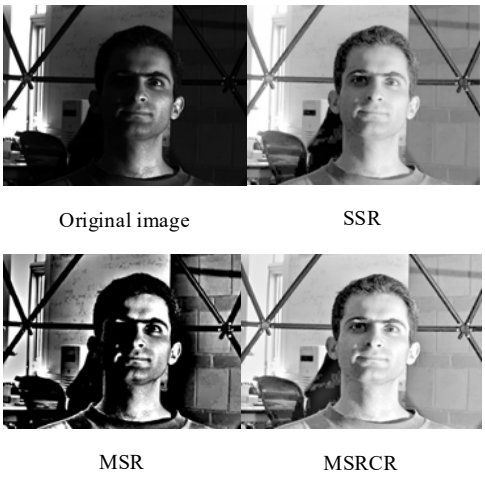

Fig. 1. Enhanced comparison of three different Retinex algorithms

3 Network Model

3.1 Introduction to the MobileNetV1

With the boom in deep learning, there has been a proliferation of convolutional neural network models in the field of computer vision. From LeNet [10] in 1998 to AlexNet, which ignited the deep learning boom in 2012, and later VGG [11] in 2014 and ResNet [12] in 2015, deep learning network models have been used increasingly well in image processing. However, with the continuous expansion of the neural network scale, the structure of the system becomes more and more complex. Thus, running deep learning neural network models requires servers with high computing power. In April 2017, Howard [13] et al. proposed a small and efficient convolutional neural network model called the MobileNetV1 network. It introduces a convolutional form of depthwise separable convolution (DSC) and optimizes ordinary convolution. Standard convolution is the combination of a set of convolution kernels with the input data to form the output of a single channel feature. However, DSC decomposes the standard convolution operation into depthwise convolution and pointwise convolution. Depthwise convolution is the application of a single fixed-size convolutional kernel to each channel of the feature map, with an equal number of kernels and channels. Pointwise convolution is the fusion of channel information by collecting feature information from each point with 1×1 convolutional kernels.

The computation of standard convolution is as:

$$K \cdot K \cdot M \cdot N \cdot F \cdot F. \tag{7}$$

The computation of DSC is:

$$K \cdot K \cdot M \cdot F \cdot F + M \cdot N \cdot F \cdot F. \tag{8}$$

The comparison between DSC and standard convolution computation is as follows:

$$\frac{K \cdot K \cdot M \cdot F \cdot F + M \cdot N \cdot F \cdot F}{K \cdot K \cdot M \cdot N \cdot F \cdot F} = \frac{1}{N} + \frac{1}{K^2}. \tag{9}$$

Here F represents the width and height of the input feature matrix (assuming equal width and height), K is the size of the convolution kernel, M is the number of channels in the input feature matrix, and N is the number of channels in the output feature matrix. In equation (9), assuming a step size of 1, if a convolution kernel of size 3×3 is used, the ordinary convolution is theoretically 8 to 9 times more computationally intensive than the DSC. It can be seen that the DSC significantly reduces the number of parameters and computations.

The basic structure of the DSC is shown in Fig. 2. We can see that the DSC convolution uses the ReLU activation function after both depth convolution and point convolution. However, during deep convolution, the depth convolution has no ability to change the number of channels, the extracted features are single-channel [14]. As a result, the ReLU activation function can cause information loss when operating on the output of convolutional layers with fewer channels.

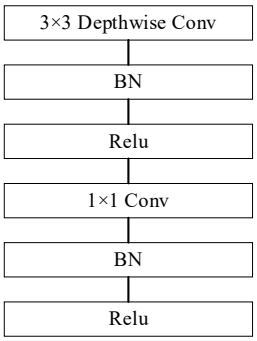

Fig. 2. DSC layer in MobileNetV1

3.2 Attention Module

Different illumination changes have a bad influence on the recognition accuracy. When the MobileNetV1 network extracts features, we can pay more attention to facial features, so we introduce an attention module into the MobileNetV1 network model. CBAM [15] is a lightweight attention module designed for convolutional neural network. It could concentrate on the channel and spatial dimensions. CBAM can be embedded in a convolutional neural network, which increases the performance of the network by a few computations. The basic structure of CBAM is shown in Fig. 3.

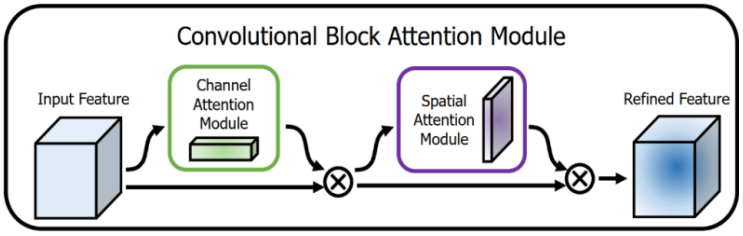

Fig. 3. The overview of convolution block attention (CBAM) [15]

CAM specific calculation process: Assume that the dimensions of the input feature map F are $H \times W \times C$. H, W and C are the height, width and channels of the feature map respectively. CAM puts the input feature map F through global maximum pooling and global average pooling based on width and height, respectively, to obtain two $1 \times 1 \times C$ feature maps. Next, they are fed into a two-layer multilayer perceptron (MLP) with the first layer of C/r neurons (r is the reduction rate) and an activation function of ReLU, and a second layer of C. The output features of the MLP are then subjected to an element-based summation operation and a sigmoid activation operation to generate the final channel attention feature map M_c. Finally, M_c and the input features F perform an elementwise multiplication operation to obtain the input features F' needed for SAM. The formula for the CAM module is shown in Eq. (10).

$$\begin{aligned} M_c(F) &= \varphi\left(MLP\left(AvgPool\left(F\right)\right) + MLP\left(MaxPool\left(F\right)\right)\right) \\ &= \varphi(w_1(w_0 F^c_{Avg}) + w_1(w_0 F^c_{Max})). \end{aligned} \tag{10}$$

Here, φ represents the sigmoid activation function, F denotes the input feature map, F_{Avg}^{c} denotes the features obtained after global average pooling of the feature map F, and F_{Max}^{c} denotes the features obtained after global maximum pooling of the feature map F. w_0 and ω_1 represent the two-layer parameters of the MLP.

SAM specific calculation process: The output feature map of CAM is taken as the input feature map of SAM. First, to obtain two $H \times W \times 1$ feature maps, the SAM module performs a channel-based maximum pooling and global averaging operation. And then performs a channel-based channel stitching operation on these two feature maps. Then, a 7×7 convolution operation is performed to reduce the dimensionality to 1 channel. The spatial attention feature map M_s is generated by the sigmoid function. Finally, the feature map is multiplied by the input feature map of the module to obtain the final generated features. The formula for the SAM module is shown in Eq. (11).

$$M_s(F) = \varphi\left(f^{7\times7}\left(\left[\left(AvgPool(F); MaxPool(F)\right]\right)\right)\right) = \varphi(f^{7\times7}([F_{Avg}^{s}; F_{Max}^{s}]). \quad \textbf{(11)}$$

Where φ represents the sigmoid activation function, F_{Avg}^{s} and F_{Max}^{s} denote the average pool feature and the maximum pool feature in the channel respectively. $f^{7\times7}$ denotes a convolution kernel of size 7×7 which is the spatial feature extractor. In this paper, different convolutional kernels will be used for experiments and comparisons. For the two modules in CBAM, we use a sequential arrangement of the two modules embedded in the MobileNetV1 network, as shown in Fig. 4.

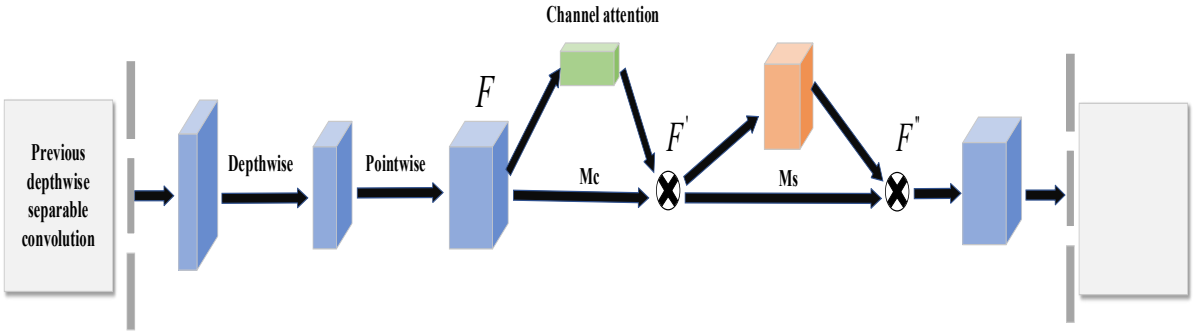

Fig. 4. CBAM embedded with a DSC Block in MobileNetV1

3.3 Arcface

In 2016, Wen et al [16] proposed L-softmax at ICML, opening the way for face recognition to improve on softmax loss. Moving from the Euclidean distance space of features to the angular space increases inter-class differentiation and intra-class cohesiveness by adding an angular distance (Margin) and expanding the critical interval. However, when using L-softmax, L-softmax needs to determine a center for each class. This is equivalent to adding the calculation and updating of the category center while performing normal classification, which requires higher hardware. Further, SphereFace [17] proposed A-softmax, which normalises the weight vector and completely angular space constraints to a hypersphere, making the classification judgement more discriminative and accurate. However, the computation of its loss function requires a series of approximations, which leads to instability in network training. L2-Softmax [18], on the other hand, performs L2 normalisation of the feature vector, further strengthening the hypersphere. It was not until ArcFace [19], presented at CVPR 2019, combined the two and further improved the angular distance, making training easier to converge.

The softmax loss only guarantees good separability between classes when the learned features do not have to do any metric learning, which can make the face features separable. However, softmax loss does not guarantee that the features learned by positive pairs are close enough and the features learned by negative pairs are far enough. To maximise highly distinguishable features for face recognition, ArcFace incorporates the margin into the established loss function, maximises the classification boundaries directly in angular space, and optimises the

geodesic distance margin directly by normalising the exact correspondence between angles and arcs in the hypersphere. Therefore, I-MobileNet uses Arcface as the classifier. The ArcFace loss function is as follows:

$$L_{Arcface} = -\frac{1}{M}\sum_{i=1}^{M}\log\left(\frac{e^{s(\cos(\vartheta_{y_i}+t))}}{e^{s(\cos(\vartheta_{y_i}+t))}+\sum_{j=1,j\neq y_i}^{m} e^{s\cos\vartheta_j}}\right). \tag{12}$$

Here, M is the size of the batch size, m is the number of categories, t is the corner margin and s is the scaling radius.

4 I-MobileNet Network Structure

In the MobileNet network, the ReLU activation function is used in the DSC layer. However, the ReLU activation function has many drawbacks. If the network has a huge gradient flowing through a ReLU neuron during forward propagation, the neuron will no longer have an activation function after the network has updated its parameters. The gradient of neuron will always be zero. In this case, the weights cannot be updated and the network cannot learn, which may lead to information loss. Therefore, we further improve the DSC layer to use linear output after depthwise convolution. The improved DSC layer is shown in Fig. 5. The computational effort of the improved DSC layer is the same as before, retaining the advantage of the reduced convolutional computation of DSC in the MobileNetV1 network. At the same time, the improved DSC layer uses a linear output after the depth convolutional layer, so that the information of each channel is fully preserved.

The attention mechanism allows the model to focus on the important information. Based on the characteristics of the attention mechanism, we concatenate the channel attention mechanism and the spatial attention mechanism in the CBAM module in turn after the point convolution of the improved DSC layer. This allows the model to focus on key features and ignore dispensable features in both the channel and spatial dimensions.

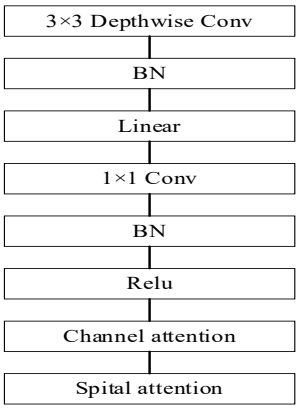

Fig. 5. The improved DSC layer

For the choice of classifier, the MobileNeV1 classifier chooses the most traditional softmax. For the I-MobileNet network, the Arcface classification is better than the softmax classification. Therefore, we choose the Arcface classifier to replace the original softmax classifier, making the intra-class spacing smaller and the extra-class spacing larger. Thus, the model can get better recognition results. The structure of the I-MobileNet is shown in Table 1.

The pre-processed face image is fed into the model, which passes through a standard convolutional layer, then sequentially through 11 DSC layers and 2 improved DSC layers embedded with CBAM, and finally through an AvgPool layer and a fully-connected layer.

Table 1. I-mobilenet network structure

Type/Stride	Filter shape	Input size
Conv/s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw/s1	3 × 3 × 32 dw	112 × 112 × 32
Conv/s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw/s2	3 × 3 × 64 dw	112 × 112 × 64
Conv/s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw/s1	3 × 3 × 128 dw	56 × 56 × 128
Conv/s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw/s2	3 × 3 × 128 dw	56 × 56 × 128
Conv /s1	× 3 × 128 × 256	28 × 28 × 128
cbam channel attention		28 × 28 × 256
spatial attention		28 × 28 × 256
Conv dw/s1	3 × 3 × 256 dw	28 × 28 × 256
Conv/s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw/s2	3 3 × 256 dw	28 × 28 × 256
Conv/s1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Conv dw/s1	3 3 × 512 dw	14 × 14 × 512
Conv/s1	1 × 1 × 512 × 512	14 × 14 × 512
cbam channel attention		14 × 14 × 512
spatial attention		14 × 14 × 512
Conv dw/s2	3 × 3 × 512 dw	14 × 14 × 512
Conv/s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw/s1	3 × 3 × 1024 dw	7 × 7 × 1024
Conv/s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool/s1	Pool 7 × 7	7 × 7 × 1024
FC/s1	1024 × 7	1 × 1 × 1024
Arcface Loss	classifier	1 × 1 × 512

5 Experiment

5.1 Experiment Preparation

Experimental environment: PyTorch is a deep learning library for applications such as image processing. Experiment was conducted on a Windows 10 64-bit operating system using the programming language Python 3.6 and the framework platform PyCharm2021.2. GPU using NVDIA 2060 with 16 GB of RAM. The dataset used the Extended Yale B dataset. In the experiment, the losses were optimised using the Adam optimiser, setting the learning rate to 0.01, the decay to 1-2, the epoch to 40 and the batch size to 80.

Experimental datasets: This paper uses the Extended Yale B dataset. The Extended Yale B dataset has 16128 facial images, collected from 38 subjects in 9 poses and 64 lighting conditions. This dataset contains facial images under various lighting conditions and can be used as a dataset for face recognition under non-uniform lighting.

Face image illumination pre-processing: In face recognition, feature extraction is a key aspect, and the expressiveness of the extracted features has a great impact on the recognition accuracy. In real scenarios, non-uniform illumination makes the extraction of face features more difficult. Therefore, image pre-processing is required before training. In this paper, we used the MSRCR algorithm to pre-process the face images of the Extended Yale B dataset for illumination to reduce the effect of illumination and highlight facial features.

5.2 Experiments on the Dataset

The network model designed in Table 1 is trained. We fed the pre-processed face images into the network to evaluate the network performance using the receiver operating characteristic curve (ROC curve). The training of the model was first carried out, and the cross-entropy loss function was used for training. The cross-entropy loss function is commonly used in classification problems. In order to calculate the loss of the network, the output of the model is ensured to be normalised between 0 and 1.

The cross-entropy describes the distance between two probability distributions, and as the cross-entropy decreases, the closer the two probability distributions become. The loss function curve of the training set is shown in Fig. 6. The results yield that the loss value gradually decreases with increasing number of iterations until it approaches 0.

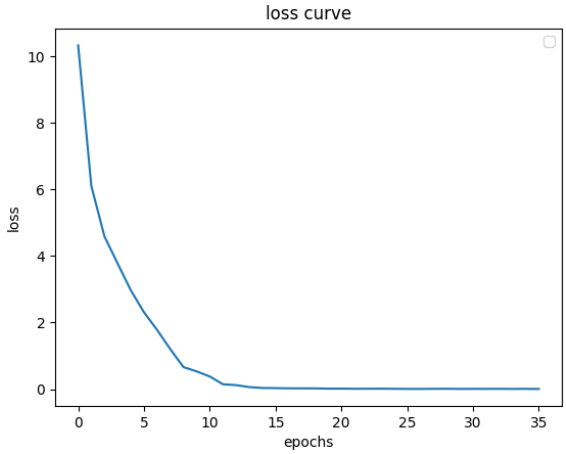

Fig. 6. Loss curve for training set

In the test set experiments, this paper performs positive and negative pairings on the extracted dataset face images. Faces of the same class are positive sample pairs, and faces of different classes are negative sample pairs. We use the ROC curve to objectively evaluate the proposed model. The ROC curve is a plot of the coordinates consisting of the False positive rate (FPR) on the horizontal axis and the True positive rate (TPR) on the vertical axis, and the different results obtained by the subject under a given stimulus condition due to the use of different judgement criteria. A common feature of traditional methods of evaluating diagnostic tests is that the test results must be divided into two types and then statistically analysis. Different from traditional evaluation methods, the ROC curve evaluation method does not need this restricted condition. The ROC curve can be used to judge whether the algorithm is good or not. The criterion is that the curve closer to the top left corner represents a better algorithm.

The ROC curve for this model on the Extended Yale B datasets is shown in Fig. 7. Here, the orange and red curves represent the results for positive and negative sample pairs on the Extended Yale B dataset respectively. It can be seen that the algorithm in this paper performs better on the Extended Yale B dataset, and the effect of positive samples on the ROC curve is better than that of negative samples.

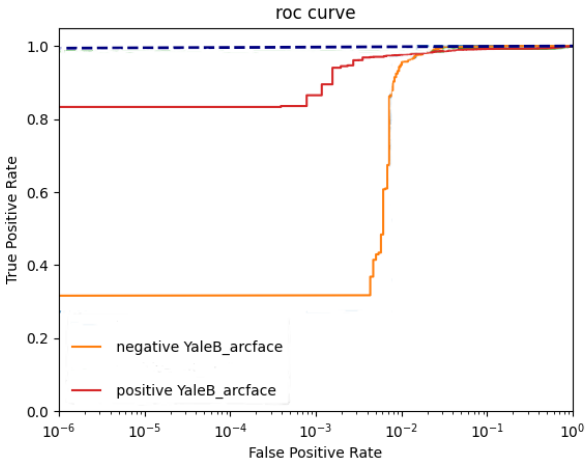

Fig. 7. ROC curve

In this paper, we compare the influence of CBAM spatial module dimensions (3×3 , 5×5 , 7×7) on the structural accuracy of i-mobilenet network, as shown in Table 2. The experimental results show that the effect is best in dimension 7×7 , so the CBAM space module dimension of 7×7 is chosen in this paper.

Table 2. The effect of spatial module dimension size on the accuracy of the I-MobileNet network

Space channel	Accuracy (%)
3×3	98.57
5×5	98.74
7×7	99.14

With the same parameter settings and convolutional layer settings, this paper compares the accuracy of I-MobileNet, MobileNetV1 network models and other models on the Extended Yale B dataset. As shown in Table 3. It can be seen that the accuracy of the I-MobileNet model in this paper is higher than that of the original MobileNetV1 model, indicating that the insertion of the CBAM module in MobileNetV1 has a significant accuracy improvement. Compared with other models, the model in this paper has better recognition performance.

Table 3. Performance evaluation on AR and Extended Yale B datasets

Model	Accuracy (%)	Parameters
N. P. Ramaiah [5]	94.01	-
IN-Net + VGG [6]	91.82	-
Y. T. Han [20]	91.3	-
VGG16	65.69	-
MobileNetV1	98.14	3.2M
I-MobileNet	99.14	3.3M

6 Conclusion

In this paper, the I-MobileNet model is proposed for face recognition under non-uniform illumination. In the I-MobileNet model, the problem of possible information loss due to the use of a non-linear activation function for the output of deep convolution is solved by using a linear output after deep convolution. To enhance the feature extraction ability of the model, an attention module is embedded in the model. We use the Arcface classifier to classify faces and improve the accuracy of the model for face recognition. Future research work will focus on: 1) Reasonably setting the corner margin of the Arcface classifier to solve the problem that the model is difficult to train. 2) Solving the problem of poor accuracy of negative sample pairs. 3) Optimising and improving the algorithm and trying to fuse it with other networks to enhance the extraction of face features.

7 Acknowledgement

This work was supported in part by the Sichuan Science and Technology Program of China (2022YFSY0056); in part by the Scientific Research Foundation of Sichuan University of Science and Engineering under Grant 2019RC11 and Grant 2019RC12.

References

[1] Y. Adini, Y. Moses, S. Ullman, Face Recognition: The Problem of Compensating for Changes in Illumination Direction, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7)(1997) 721-732.

[2] S.-G. Shan, W. Gao, B. Cao, D.-B. Zhao, Illumination Normalization for Robust Face Recognition Against Varying Lighting Conditions, in: Proc. 2003 IEEE International SOI Conference, 2003.

[3] Y. Taigman, M. Yang, M.-A. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[4] C. Szegedy, W. Liu, Y.-Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[5] N.-P. Ramaiah, E.-P. Ijjina, C.-K. Mohan, Illumination invariant face recognition using convolutional neural networks, in: Proc. 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015.
[6] Y.-H. Kim, H. Kim, S.-W. Kim, H.-Y. Kim, S.-J. Ko, Illumination normalisation using convolutional neural network with application to face recognition, Electronics Letters 53(6)(2017) 399-401.
[7] T.-C. Sun, J.-T. Barron, Y.-T. Tsai, Z.-X. Xu, X.-M. Yu, G. Fyffe, C. Rhemann, J. Busch, P. Debevec, R. Ramamoorthi, Single Image Portrait Relighting, ACM Transactions on Graphics (TOG) 38(4)(2019) 79:1-12.
[8] E.-H. Land, J.-J. McCann, Lightness and retinex theory, Journal of the Optical Society of America 61(1)(1971) 1-11.
[9] D.-J. Jobson, Z. Rahman, G.-A. Woodell, A multiscale retinex for bridging the gap between color images and the human observation of scenes, IEEE Transactions on Image processing 6(7)(1997) 965-976.
[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11)(1998) 2278-2324.
[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science 52(3)(2014) 1-14.
[12] K.-M. He, X.-Y. Zhang, S.-Q. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[13] A.-G. Howard, M.-L. Zhu, B. Chen, D. Kalenichenko, W.-J. W, T. Weyand, M. Andreetto, H. Adam, Mobile Nets: efficient convolutional networks for mobile vision applications. <http://arxiv.org/abs/1704.04861>, 2017 (accessed 17.04.17).
[14] W.-X. Wang, X. Zhou, X.-H. He, L.-B. Qing, Z.-Y. Wang, Facial expression recognition based on improved MobileNet network, Computer applications and software 37(4)(2020) 137-144.
[15] S. Woo, J. Park, J.-Y. Lee, I.-S. Kweon, CBAM: convolutional block attention module, in: Proc. European Conference on Computer Vision (ECCV), 2018.
[16] W.-Y. Liu, Y.-D. Wen, Z.-D. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks. <http://arxiv.org/abs/1612.02295>, 2016 (accessed 07.12.16).
[17] W.-Y. Liu, Y.-D. Wen, Z.-D. Yu, M. Li, B. Raj, L. Song, SphereFace: Deep Hypersphere Embedding for Face Recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[18] R. Ranjan, C.-D. Castillo, R. Chellappa, L2-constrained softmax loss for discriminative face verification. <http://arxiv.org/abs/1703.09507>, 2017 (accessed 28.03.17).
[19] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[20] Y.-T. Han, G.-J. Lin, L.-J. Zhao, X.-L. Tang, Y. Huang, H. Jiang, An Improved Homomorphic Filtering Algorithm for Face Image Preprocessing, Journal of Computers 32(6)(2021) 66-82.