International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n2.02
Journal homepage: https://wepub.org/index.php/IJCSIT/index

A Face Recognition Method Based on Transfer Learning and Attention Mechanism

Aodi Zhang, Shibao Sun*

Henan University of Science and Technology, Luoyang, China
*Corresponding Author: Shibao Sun

ABSTRACT

With the popularization of network technology and the development of informatization, the security of personal identity information is receiving increasing attention. In fields such as finance, healthcare, and security, security requirements are increasing, and more reliable and secure identity verification methods are needed to protect personal information from illegal acquisition and use. Traditional identity verification methods, such as passwords and PIN codes, face problems such as being guessed, forgotten, and stolen. Based on the understanding of the limitations and shortcomings of traditional identity verification methods, biometric recognition technology has emerged. At present, it mainly includes biometric recognition technologies such as fingerprint recognition[1], voice recognition[2], facial recognition[3], iris recognition, retinal recognition[4], and DNA recognition[5] . As an important branch, facial recognition technology has shown tremendous potential and advantages in the field of information security In order to study the performance of facial recognition in small sample scenarios, the ECANet attention mechanism was introduced into the classic ResNet50 network, and a new network model, ResNet50-ECA[6], was constructed. Firstly, a small-scale face SLFW dataset containing only 28 classifications was created on the famous LFW (Labeled Faces in the Wild) dataset[7]. When processing these facial images, an advanced image enhancement technique, namely adaptive histogram equalization with limited contrast, was adopted. Effectively enhances the contrast between light and dark in the image, making facial features clearer and more prominent. In order to expand the dataset and enhance the robustness of the model, data augmentation was also performed. This includes random rotation and horizontal inversion of images, which can generate more diverse and rich training samples. Pre trained network models on the ImageNet dataset with parameters and weights, then fine tuned these models and applied them to a small sample face dataset.

KEYWORDS

Facial recognition; Transfer learning; Attention mechanism; Small sample dataset

1. INTRODUCTION

This article introduces the strategy of transfer learning, adopting pre-trained network models VGG, GoogLeNet, and ResNet50 on the ImageNet dataset. These models' parameters and weights are fine-tuned and transferred, demonstrating excellent performance when applied to the target small-sample face dataset. To further enhance the model's ability to extract features, this study introduces an attention mechanism, namely the ECANet module. Building upon ResNet50, a new model ResNet50_ECANet is constructed, which more effectively focuses on the extraction and utilization of key feature information. By comparing the ResNet50_ECANet model with the introduction of the attention mechanism to other network models that do not use this mechanism, the experimental results fully validate the outstanding performance of the proposed model on the small-sample face dataset.

Its generalization ability is significantly improved, there by achieving a higher level of model recognition accuracy.

2. THEORETICAL INTRODUCTION

2.1. ResNet50 Model selection

The VGG[8], GoogLeNe[9]t, and AlexNet models are classic convolutional neural networks. The network structure of VGGNet is relatively simple, but due to the use of very small convolutional kernels (3x3) and deeper network layers, this results in a very large number of parameters for the model. This requires more computing resources and time for training and inference with VGGNet. AlexNet uses convolutional kernels of sizes 5x5 and 3x3, but overall, its receptive field is still relatively small. This means the network might overlook some larger-scale feature information, affecting the understanding of the overall image structure.

In the multiple Inception modules of GoogLeNet, the use of different scales of convolutional kernels and pooling layers might lead to redundant extraction of some features, increasing the network's redundancy.ResNet50 is a classic model in the Residual Network (ResNet) series. The network model structure of ResNet50 is shown in Figure 1-1.

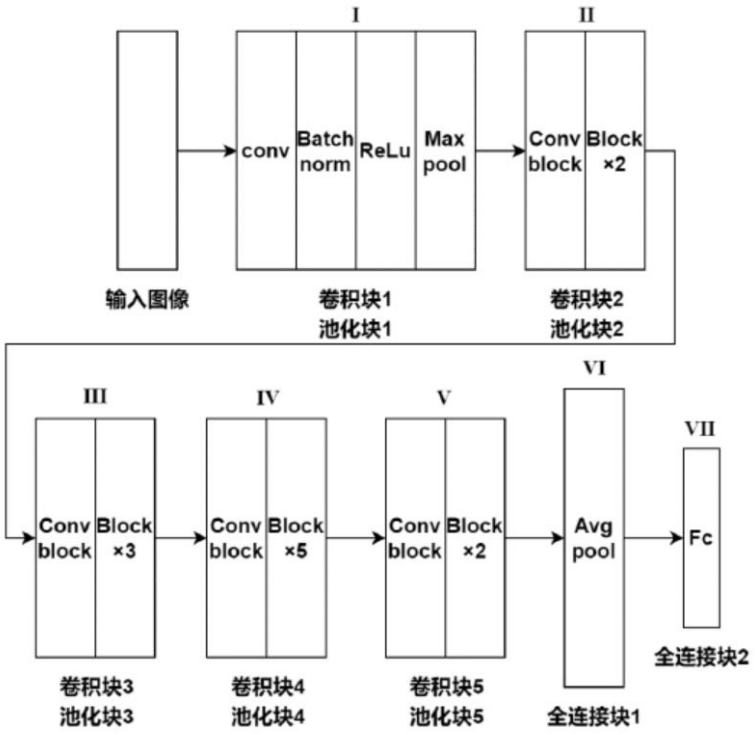

Fig. 1-1 ResNet50 Network Model Structure

As shown in the figure, the ResNet50 network model structure can be divided into 7 parts. Input layer: Accepts input image data and performs initial convolution operations.

Batch normalization layer: performs batch normalization on the output of convolutional layers ReLU activation function (Corrected Linear Unit): activates the output of convolutional layers Max Pooling: A 3x3 sized pooling kernel with a step size of 2, used for pooling operations.

Stage 2: Residual Blocks: Each residual block contains 3 convolutional layers, using 1x1, 3x3, and 1x1 convolution kernels without changing the size of the feature map. There are convolutional layers at the beginning and end of each residual block, which are used to adjust the size of the feature map and the number of channels.

Stage 3: Four residual blocks with the same structure, each containing three convolutional layers.

Stage 4: Six residual blocks with the same structure, each containing three convolutional layers.

Stage 5: Three residual blocks with the same structure, each containing three convolutional layers Global Average Pooling [36]: Perform global average pooling on the output of the last residual block, and summarize the feature map size into 1x1 Fully Connected Layer: used to map pooled feature maps to output categories, with 1000 output nodes corresponding to 1000 categories in the ImageNet dataset. The residual blocks are not included in the parts of l, Vl, and Vll in the figure. Usually, the softmax function is used for final classification. The ResNet series has wide applicability and maturity in practical applications due to its different network depths and structural variants. In the ResNet series, there are multiple versions to choose from, such as ResNet18, ResNet50, ResNet101, etc., each with its specific advantages and applicable scenarios. ResNet50 is a network model trained on an ImageNet dataset containing over one million images, which contain a large number of facial images. This enables the ResNet50 model to be applied to face image classification tasks with limited data through transfer learning methods. Due to the wide and diverse nature of the ImageNet dataset, ResNet50 learns rich feature information during pre training, making it a powerful tool for processing various visual tasks. The reasons for choosing ResNet50 as the basic network architecture in this article are as follows:

1. Stronger feature extraction capability: Compared to ResNet18, ResNet50 has a deeper network structure, which enables it to better extract and characterize complex image features. For the classification task of complex objects such as facial images, this deeper network structure can better capture the details and features in the Image.2.

2. Avoid overfitting: Compared to ResNet101, choosing ResNet50 can effectively avoid potential overfitting problems. ResNet101 has more layers and parameters, which may lead to the model performing too well on the training set in some cases, but its generalization ability decreases on the test set. ResNet50 has achieved a better balance in this Regard.3.

3. Balance between network depth and performance: ResNet50 achieves a good balance between network depth and model performance. It not only has sufficient depth to extract features of complex images, but also has high operability and trainability. This means that we can train and adjust more easily while maintaining model performance.

2.2. ECANet Channel Attention Mechanism

Attention mechanism is an important technique in deep learning[10], which mimics the behavior of human attention and allows neural networks to focus more on important parts when processing sequence data.

When processing sequence data, traditional neural network models (such as recurrent neural networks or long short-term memory networks) encode the input information at each time step and pass it on to the next time step or output layer [53]. However, this approach may lead to information loss or confusion, especially when the sequence is long or contains a large amount of information. The emergence of attention mechanism solves this problem. It allows models to assign different attention weights to different parts of the input sequence when processing each time step, so that the model can focus more on important parts, thereby improving the performance and generalization ability of the model. ECA Net (Efficient Channel Attention Network) is an attention mechanism used for image processing tasks, mainly used to extract inter channel relationships in image features to improve the performance of the model. The channel attention mechanism of ECANet aims to more effectively capture the correlations between different channels, thereby improving the representation ability of features. This mechanism aims to enhance the feature expression ability, make the network more focused on important channel information, and thereby improve the recognition performance of the model. ECANet is an improvement based on the SENet channel attention mechanism, which learns

attention weights between different channels to improve the efficiency and performance of the model. The structure and workflow of SENet are as follows: Firstly, the incentive branch of SENet performs global average pooling on the feature map.

Then, the feature map is generated through a dimensionality reduced fully connected layer, ReLU layer, dimensionality increased fully connected layer, and Sigmoid layer. Although it ensures accuracy, it also introduces a certain amount of parameters and computational complexity.

In fully connected layer operations, dimensionality reduction reduces the parameters of the model, but also disrupts the direct correspondence between channels and weights. This may lead to the loss of feature information, limiting the effectiveness and performance of attention mechanisms. ECANet proposed a local cross channel interaction method that does not require dimensionality reduction to extract dependencies between channels. The recognition accuracy is higher than SENet because it preserves the direct relationship between channels, improving the efficiency of feature information extraction and utilization. The ECANet network structure diagram is shown in fig2-2.

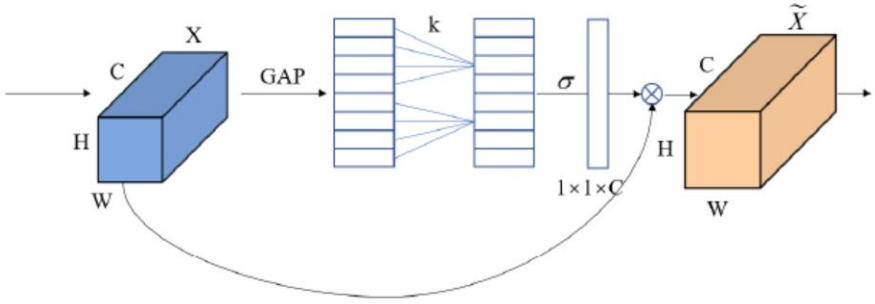

Fig. 2-2: ECANet Network Structure Diagram

Firstly, perform a global average pooling operation on features of size H x W x C to focus on the image region of interest and reduce its dimensionality. Finally, a feature representation of 1x1xC is obtained. In order to achieve information exchange between channels without the need for dimensionality reduction, a one-dimensional convolution kernel of size k is used for local convolution operations. Here, k represents the coverage during cross channel interaction, which refers to the number of surrounding elements. By activating the function σ Generate channel weights ω, Multiply the weight with the input to obtain the channel feature map. The formula for channel weight is:

$$\omega = \sigma(C1D_K(\mathrm{y})) \tag{1}$$

Here is the Sigmoid activation function, representing one-dimensional convolution, where k represents the size of the convolution kernel, and y is the 1x1xC feature being convolved. EACNet captures local cross channel information interactions by determining the range of interaction information between channels. There is a mapping relationship between the number of channels C and the size of the convolutional kernel k:

$$C = \eta(\mathrm{k}) \tag{2}$$

Generally speaking, C is an exponent of 2, but a nonlinear function, represented as:

$$C = 2^{\gamma(\eta(\mathrm{k})+\gamma^{*}-1)} \tag{3}$$

Among them , $\gamma = 2$, $\gamma^{*} = 1$, γ^{*} is related to the function C.

When the channel dimension C is determined, the size of the convolution kernel k can be calculated using a formula:

$$k = 2^{log_{odd}(b\cdot\eta(C)^{\gamma})} \tag{4}$$

$b=1$, $\gamma=2$, odd Represents the nearest odd number after taking an absolute value. The fully connected layer in SENet has been replaced by one-dimensional convolution, which helps prevent excessive model parameters and avoids information loss when converting on the channel dimension.

2.3. Build ResNet50-ECANet

In order to reduce the transmission of irrelevant feature information and improve the model's attention level in different image regions, this paper combines the ECANet channel attention mechanism with the ResNet50 network model to construct a new ResNet50-ECA model, which improves recognition accuracy on the original basis. The residual module introduced by the ECANet module into the ResNet50 network model is shown in Figures 2-3.

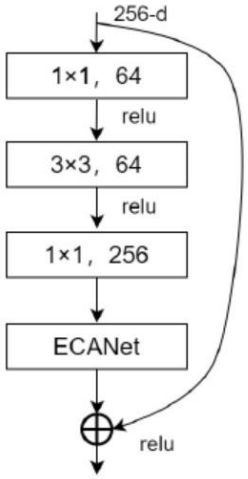

Figure 2-3: Structure of residual network with ECANet added

Each residual module incorporates an ECANet channel attention mechanism to improve the residual learning rate. ECANet utilizes fast 1D convolution operations to generate channel attention, and utilizes Global Average Pooling (GAP) to aggregate features of target related image regions, achieving local cross channel information exchange without the need for dimensionality reduction. In ECANet, the attention module determines the weight of each position on the feature map based on the content of a specific region, improving model performance. This improvement enables the model to focus more on key features, enhancing feature expression and recognition accuracy. The specific model structure is shown in Figure 2-4.

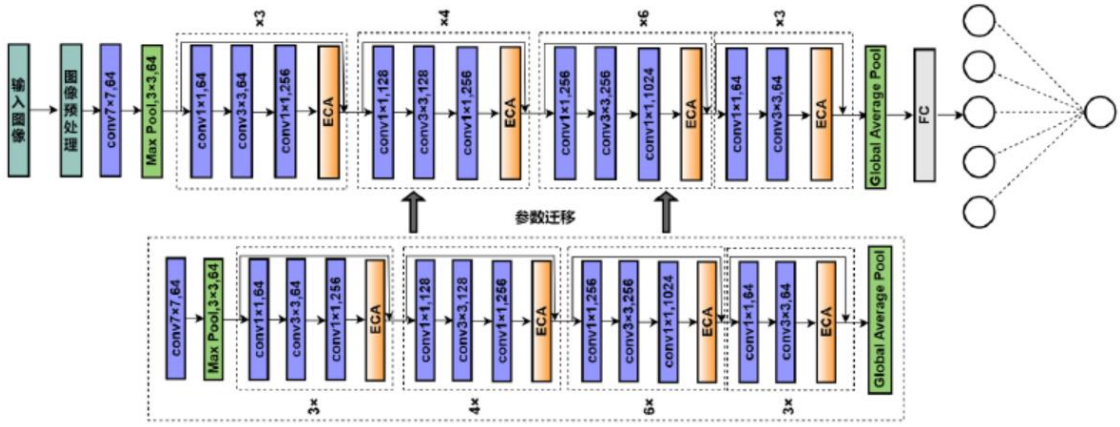

Fig. 2-4 The structure of the ResNet50_ECANet network model.

This article is a transfer learning method based on ResNet50 convolutional neural network, and the training process is shown in the following figure. Pre train the ECANet attention mechanism model

on the dataset ImageNet to obtain 10000 classification results. The entire transfer learning process involves determining network parameters such as optimizer, learning rate, and number of iterations before training begins. During the training process, the model's paranoia parameters and weights should also be continuously saved. After training is completed, these parameter files should be loaded, and the weights in the convolutional layer should be initialized. The classification model parameter settings should be transferred and kept unchanged from the source model. The weight parameters of the target dataset obtained after training are also saved.

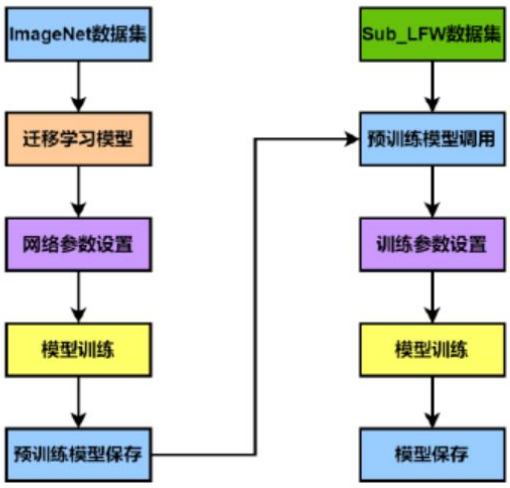

Fig. 2-5 Transfer Learning Process

2.4. Algorithm process description

Enter the facial image training dataset X, facial image test dataset Y, and iteration count E. Initialize iteration number epoch=0; Training our model ResNet50-ECANet on the ImageNet.

1. For each number in dataset X, it is used to represent.

2. For each training session, an additional 1 should be added as the size of the next session.

3. Input X into the model for feature extraction and output values.

4. Compare the output value with the true value to calculate the cross entropy loss function.

5. Use error backpropagation and update model weight 6 using Adam optimization algorithm. Repeat steps 2-5 until e=E 7. Save the obtained model 9. Input the facial image dataset Y into the saved model to obtain the predicted category of the test samples.

3. EXPERIMENT AND ANALYSIS

Firstly, we will introduce the small sample face dataset used. Afterwards, describe the hyperparameter settings used in the experiment. Finally, experimental preprocessing was conducted to compare the proposed method with other deep learning models, mainly including image enhancement processing and the introduction of attention mechanisms.

3.1. Experimental Dataset

The LFW (Labeled Faces in the Wild) face dataset is a standard dataset widely used for face recognition and related tasks. It consists of face images collected from the internet, containing faces in various real-world environments with different poses, expressions, and lighting conditions. The LFW dataset comprises over 13,000 face images from 5,749 individuals, all of which have been manually labeled with the identities of the people. These images were captured in different

environments, including indoor, outdoor, varying lighting conditions, and more. This makes the LFW dataset one of the important benchmarks for evaluating the robustness and performance of face recognition algorithms in real-world scenarios.

Each character selected in this experiment has 20-90 facial images, totaling 1100 images and 28 people, which are divided into 28 categories and form a sub dataset named S-LFW. Among them, the training set accounts for 80% of the sub dataset images, which is 880 images, and the test set accounts for 20% of the sub dataset images, which is 220 images. The images are cropped into 224 x 224 pixels in size.

3.2. Experimental Configuration

This experiment is mainly run on the Windows platform, and the specific experimental environment and computer configuration information are shown in Table 3-1.

Tab. 3-1 Experimental Data Parameters

Allocation	Specific information
CPU	Intel(R)Core(TM) I7-13700H 5.4GHz
Memory	16G×2 DDR5 6800Hz
OS	Windows11 pro
framework	PyTorch
Developing software	PyCharm 2022
CUDA	11.3

3.3. Experimental hyperparameter settings

Firstly, set the hyperparameters for the experiment before proceeding with the experiment. In order to apply the 28 person face classification on the S1LFW dataset used in this article, and initialize the network parameters with the weights of pre trained network models such as VGG, GoogLeNet, and ResNet50 in ImageNet, the number of fully connected layer classifications is modified. Divide the pre trained ImageNet model into 1000 categories and modify the number of categories to 28. The specific parameter settings are shown in Table 3-2.

Table 3-2 Parameter settings for the experiment

allocation	Parameter values	name
Learning rate	10^{-3}	Initial learning rate
optimizer	Adam	optimizer
epoch	100	Training iterations
Weight decay	0.0005	Weight attenuation value
Batch size	64	Iteration batch includes sample quantity

3.4. Experimental preprocessing

Before training a network model, image enhancement preprocessing of the face dataset is required. This refers to a series of operations performed on the face image before it is used to train a deep learning model, with the aim of enhancing the quality, diversity, and generalization ability of the image. Histogram equalization is a method used to enhance image contrast. In image processing, its main function is to improve the visual quality of images, making them clearer and with better visual effects.

Firstly, calculate the grayscale histogram of the image to obtain the frequency of each pixel value. Assuming the pixel value range of an image is 0 to L-1 (usually 0 to 255, corresponding to an 8-bit grayscale image).

Next, calculate the cumulative distribution function (CDF) of the image, representing the cumulative frequency of each pixel value. CDF can be calculated using the y function using the following formula:

$$y(i) = \sum_{j=0}^{i} \frac{n}{N} \tag{1}$$

Among them, n_j is the number of pixels with a pixel value of j, N is the total number of pixels in an image, Then apply the cumulative distribution function $y(i)$ Normalize to Range [0, L-1], Obtain new pixel values after histogram equalization h(i):

$$h(i) = \eta(L-1) \times \text{y（i）} \tag{2}$$

$\eta()$ Indicates rounding operation. Finally, map each pixel value x in the original image to a new pixel value z, applying a histogram equalization transformation:

$$z = h(x) \tag{3}$$

Among them, x It is the pixel value of the original image, z It is the pixel value after histogram equalization.

These steps process the pixel values of the original image through histogram equalization, making the grayscale distribution of the image more uniform, thereby enhancing the contrast and visual effect of the image. However, histogram equalization may result in excessive enhancement in some cases, leading to the loss of local details in the image or problems with overexposure or low contrast. This is because histogram equalization is a global pixel value transformation. Applying the same transformation to the entire image may cause certain areas of the image to become too bright or too dark, thereby affecting the visual quality and detail information of the image. To solve this problem, the Adaptive Histogram Equalization (AHE) algorithm can be used. The AHE algorithm can perform histogram equalization based on local regions of the image, enhancing the contrast of each local region and preserving more detailed information.

The basic idea of adaptive histogram equalization is to divide the image into many small local regions and apply histogram equalization to each local region separately, rather than processing the entire image globally. The AHE algorithm needs to perform histogram equalization on each local area of the image, which results in high computational complexity of the algorithm. At the same time, it will enhance the local contrast of the image and in some cases, may amplify the noise and other situations present in the image.

This article employs Contrast Limited Adaptive Histogram Equalization (CLAHE) for enhancing face images. By introducing a mechanism for limiting contrast, the CLAHE algorithm effectively addresses the issues of excessive enhancement and noise amplification that may arise with the AHE algorithm. It can enhance image contrast while preserving the natural appearance and detail information of the image. The steps of the CLAHE algorithm are as follows:

1. Divide the image into small blocks of size M×N, with each block representing a local region.

2. Perform histogram equalization on each local region to obtain the enhanced local histogram. This step is similar to the AHE algorithm.

Assuming the original image is f(x,y), Local areas are $f_i(x, y)$, the formula for local histogram equalization is shown in

$$g_i(x,y) = \frac{L-1}{MN} \sum_{k=0}^{f_i(x,y)} n_i(k) \tag{4}$$

L is the number of grayscale levels of pixel values, MN is the total number of pixels in a local area, $n_i(k)$ is the number of pixels with a pixel value of k in the local area.

3. To limit contrast, the enhanced local histograms of each local area are cropped. The specific approach is to calculate the cumulative distribution function (CDF) of the local histogram and limit it to a maximum value T. If the number of pixels in a certain grayscale level exceeds the threshold T, these pixels are evenly distributed to other grayscale levels. The formula is shown in (5).

$$\begin{aligned} &if\ \ g_i(x,y) \le T \rightarrow g_i'(x,y) = g_i(x,y) \\ &if\ \ g_i(x,y) \succ T \rightarrow g_i'(x,y) = T \end{aligned} \tag{5}$$

4. Finally, crop the local histogram $g_i(x,y)$ interpolate or concatenate to obtain the CLAHE result of the entire image, as shown in the formula (6).

$$\hat{f}(x,y) = \frac{L-1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} g'_i(x,y) \omega_i(x,y) \tag{6}$$

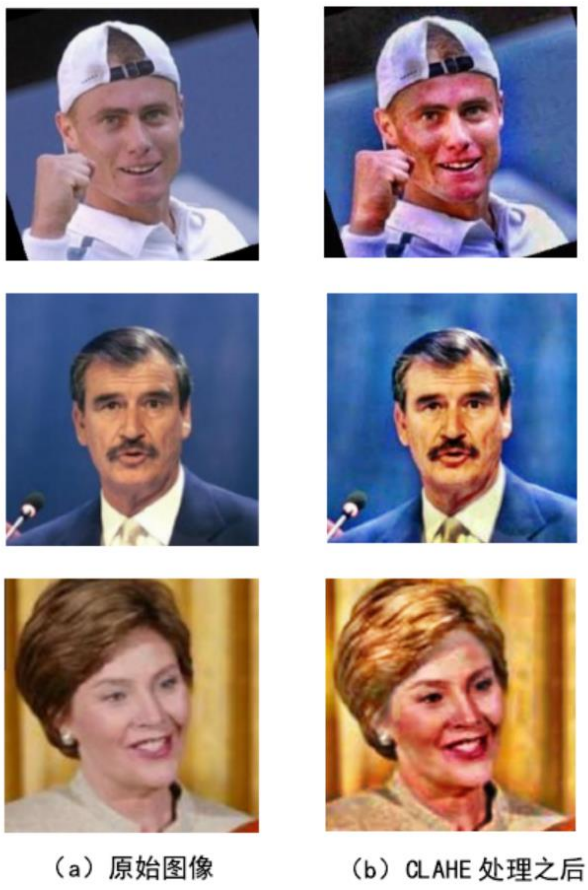

Fig. 3-1 CLAHE Facial Image Enhancement Processing

CLAHE (Contrast Limited Adaptive Histogram Equalization) is an effective technique for processing facial images. From Figure 3-1 (a), it can be seen that the original facial image often lacks detailed information and contrast in terms of brightness and darkness. However, after CLAHE processing, Figure 3-1 (b) of the facial image shows a significantly enhanced light dark relationship, and the facial detail features are more clearly visible In order to further improve the generalization ability of the

face classification network model, various image enhancement techniques were used in the experiment, and experiments were conducted on datasets such as random rotation, cropping, and horizontal reversal. These technologies not only increase the diversity of the S1LFW facial image dataset, but also expand the size of the dataset, which is beneficial for improving the performance of the network model. In the image processing stage, the PyTorch framework's ToTensor function was used to convert the image into the Tensor format required by the network model. This step normalizes pixel values to the range of [0.0, 1.0], which helps improve the stability and convergence speed of model training. In addition, the images were standardized by using the Normalize function to make the data more consistent with the centralized distribution pattern. This step further enhances the model's generalization ability and training effectiveness In summary, the aim of the experiment is to improve the recognition and generalization performance of the network model for facial features through CLAHE processing of facial images, the use of various image enhancement techniques, and the combination of PyTorch framework processing methods, ultimately improving the accuracy and stability of facial classification.

3.5. Experimental results processed by CLAHE

In order to verify the model improvement effect after processing facial images with CLAHE, experiments were conducted on three classic network models, ResNet50, VGG, and GoogLeNet, with the hyperparameters set above, on the sub dataset S-LFW. The experimental results are shown in Table 3-3.

Tab. 3-3 Accuracy of Three Models on the S-LFW Dataset

Model	Train Accuracy	Test set accuracy
GoogLeNet	92.48%	91.73%
GoogLeNet_S	99.71%	97.21%
ResNet50	96.14%	95.98%
ResNet50_S	97.51%	96.24%
VGG	92.11%	91.01%
VGG_S	99.91%	92.99%

GoogLeNet_S, ResNet50_S, and VGG_S represent the images processed with CLAHE input into their respective models for training and testing. After enhancing the face images with CLAHE, the accuracy of GoogLeNet_S is 97.21%, ResNet50_S is 96.24% and VGG_S is 91.99%. Compared to the three models without CLAHE processing, their accuracies have improved by 5.48%, 0.26% and 1.98%, respectively. It can be concluded that the network models constructed with the introduction of the CLAHE method can enhance the brightness and contrast of face images, strengthen facial feature details, and effectively improve accuracy.

3.6. Network model experiment with ECANet module added

In order to further improve the accuracy of facial recognition in the network model, the channel attention mechanism ECANet module and SENet module were added on the basis of ResNet50, forming the ResNet50-ECA and ResNet50-SE network models. Firstly, use the convolutional module of ResNet50 to extract facial features from the S1LFW dataset. Then, the ECANet and SENet channel attention mechanism modules are used to learn these feature maps separately, and important feature information is processed by increasing attention weights. Finally, the classification results are output through the fully connected layer. The hyperparameters of the model set below will be tested on the S1LFW face dataset according to the settings in section 4.6.3. The results are shown in Table 3-4.

Table 3-4 ResNet model accuracy on dataset S1LFW

Module	Train correct rate	Test correct rate
ResNet50	96.14%	95.98%
ResNet50_ECA	99.51%	98.24%
ResNet50_SE	98.11%	97.15%

As shown in Table 3-4, the two models ResNet50-ECA and ResNet50-ECA, which introduce channel attention mechanisms ECANet and SENet modules on the basis of the ResNet50 model, achieved good results of 98.24% and 97.15% on the dataset, improving the accuracy by 2.26% and 1.17% compared to the original ResNet50. It can be seen that introducing channel attention mechanisms can significantly improve the recognition accuracy of the network model in facial images. The reason why the accuracy of the ResNet50-ECA network model is higher than that of the ResNet50-SE network model. The main reason is that the SENet module reduces the number of parameters when performing dimensionality reduction operations on fully connected layers, but it disrupts the direct correspondence between channels and weights, resulting in the loss of feature information and affecting the accuracy of the network model. In contrast, the ECANet module adaptively learns the weights of each channel and weights important channels for fusion, highlighting image features and thus improving the accuracy of facial recognition in the network model.

In order to further analyze the performance of the models, the following experiments will be conducted to compare the performance of each network model on the same dataset: comparative testing experiments of four models, ResNet50-ECA, ResNet50, VGG, and GoogLeNet, on the S1LFW face dataset. Save the best training model every 5 iterations and end after 100 iterations.

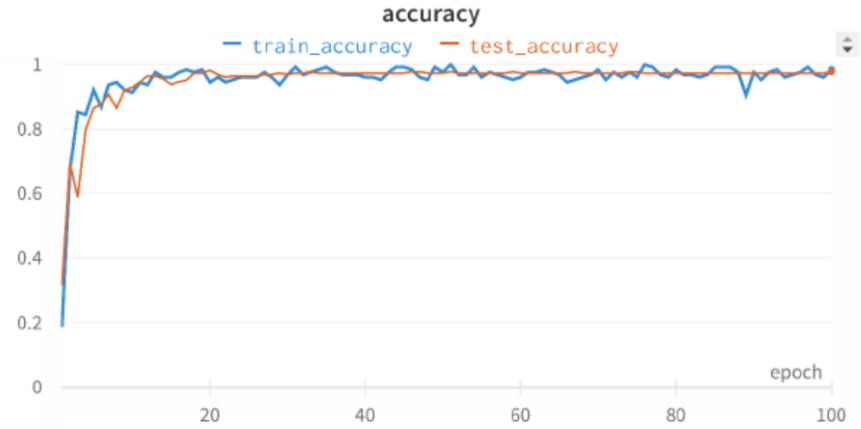

Fig. 3-2 Experimental results of ResN50-ECA model

The variation of training and testing accuracy of the ResN50-ECA model on the S1LFW dataset as the number of iterations increases, as shown in Figure 3-2. After training the network model for about 30 iterations, the accuracy of the test set tends to converge, with a test set accuracy of 98.24%.

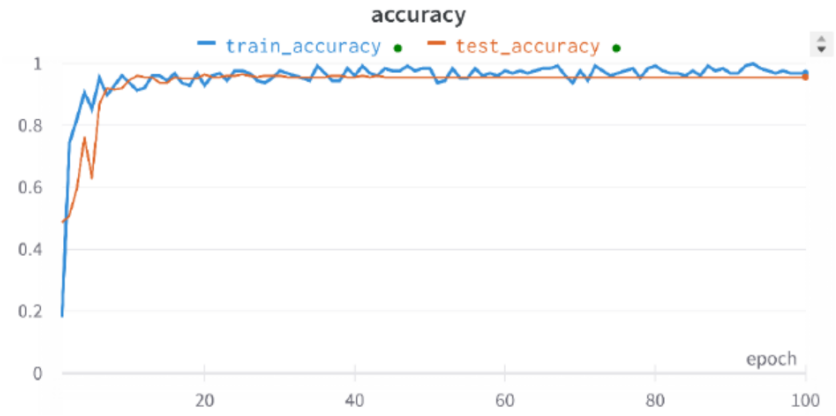

Figure 3-3 Experimental results of ResN50 model

Figure 3-3 shows the variation of training and testing accuracy of the ResNet50 network model on the S1LFW face dataset with increasing iteration times. After approximately 40 iterations of training, the accuracy of the test set tends to converge without significant changes, and the accuracy of the test set is 95.98%.

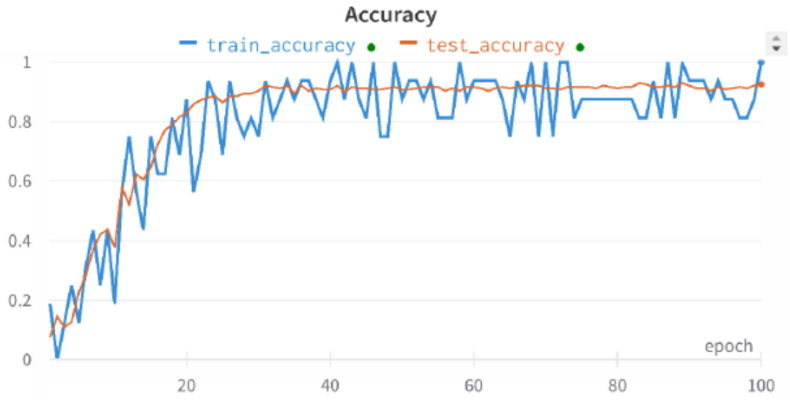

Fig. 3-4 Experimental results of VGG model

Figure 3-4 shows the changes in training and testing accuracy of the VGG network model on the S1LFW face dataset with increasing iterations. Although the model fluctuated significantly in the early stages, its accuracy on the test set stabilized after about 60 iterations of training, and remained at 91.01%.

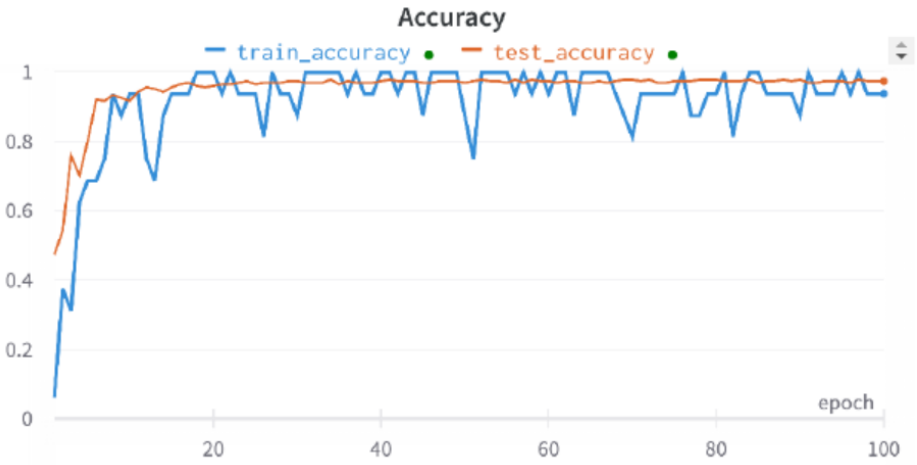

Fig. 3-5 Experimental results of the GoogLeNet model.

Figure 3-5 shows the variation of the training and testing accuracy of the GoogLeNet network model on the S1LFW face dataset with increasing iterations. After approximately 60 iterations of training, the accuracy on the test set tends to stabilize and converge, and the accuracy remains at 97.13%.

According to Figures 3-6, it can be seen that as the number of iterations increases, the loss function curve continuously decreases, indicating that the model tends to converge. The proposed ResN50-ECA model method that introduces attention wit tends to stabilize the loss function at around 40 epochs, and the convergence speed is much faster than other models. The loss graphs of each model on the training set are shown in Figure 3-6.

Fig. 3-6 Training loss diagrams for four models

Based on the above experimental results, ResNet50_ECA, ResNet50, VGG, GoogLeNet as shown in Table 3-5.

Tab. 2-5 Experimental results of four models on SLFW

Model	Accuracy rate	F1	Recall rate
ResNet50	95.98%	0.9622	0.9605
ResNet50_ECA	98.24%	0.9777	0.9802
VGG	91.01%	0.9176	0.9213
GoogLeNet	97.13%	0.9718	0.9753

The F1 value in Tables 3-5 represents the f1 value. From the table, it can be seen that the ResNet50-ECA network model with the ECANet module has a relative accuracy of 98.24%, a recall rate of 0.9802, and a f1 value of 0.9802. Compared with the other three models, the ResNet50-ECA network model with the ECANet module is superior in terms of recall rate, f1 value, and experimental accuracy compared to the network model without the ECANet module.

As shown in Figures 3-7 and 3-8, the proposed method has reduced the running time by 2.2% compared to ResNet50, 38.1% compared to the GoogLeNet network model, and 48.1% compared to the VGG model, indicating a significant improvement. Since the method proposed in this article consists of a global average pooling layer and two fully connected layers to form the ECANet module, and their parameters are independent of the input feature map, the size of the model will not be increased. This mechanism adaptively learns the weights of each channel, avoiding unnecessary calculations in the model and reducing the running time of the model. Compared with GoogLeNet, the method proposed in this article has less runtime.

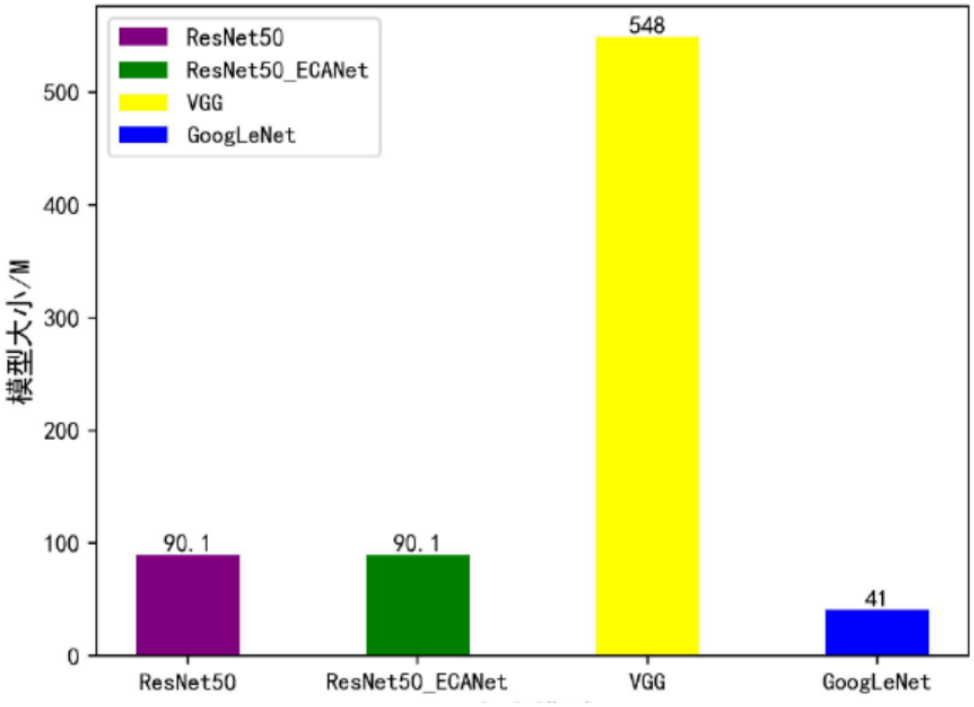

Fig. 3-7 Size of each model experiment

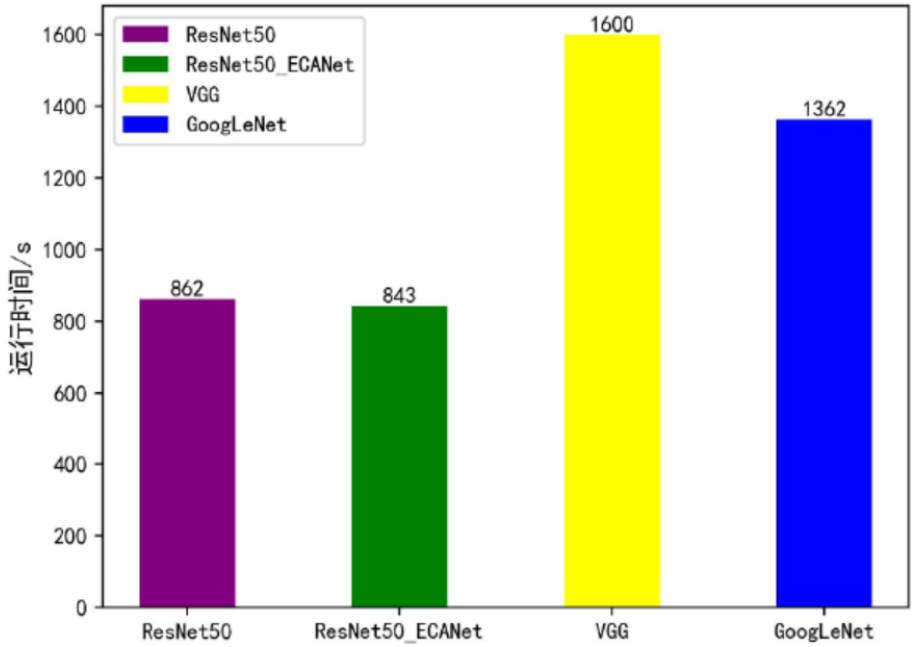

Fig. 3-8 Running time of various model experiments

In summary, for small sample face datasets, the method proposed in this paper improves face recognition accuracy by 1.81% to 5.92% and requires 2.2% to 48.1% less time than other deep learning models. Therefore, it can be said that the model has a significant improvement in facial recognition performance.

4. SUMMARY

This chapter adopts transfer learning method to transfer the pre trained ResNet50 network model parameters and weights on the ImageNet dataset, and applies them on the self built small sample face dataset S1LFW. This avoids training the network model from scratch, greatly accelerates and optimizes the learning efficiency of the network, and saves computational resources. In order to enhance the generalization ability of the model, we used data augmentation methods such as random rotation, cropping, and horizontal reversal to expand the dataset before the experiment. In addition, we introduced adaptive histogram equalization with limited contrast to enhance the brightness relationship of the input facial image, thereby enhancing the detailed information of the face. Furthermore, we integrated the ECANet channel attention module into the ResNet50 network model to form the ResNet50-ECA network model, effectively improving the recognition accuracy of the model. Through comparative experiments with other pre trained network models, we found that the ResNet50-ECA model has the best performance. We also validated the feasibility of the algorithm based on the fusion of transfer learning and attention mechanism, demonstrating good performance on small sample face datasets.

ACKNOWLEDGEMENTS

His work was supported by the Longmen laboratory for Exploratory Research Project. [No. MQYTSKT034].

This work was supported by the Longmen laboratory for Exploratory Research Project. [No. MQYTSKT034].

REFERENCES

[1] MARASCO E, ROSS A J A C S. A survey on antispoofing schemes for fingerprint recognition systems [J]. 2014, 47(2): 1-36.

[2] ANDICS A, MCQUEEN J M, PETERSSON K M, et al. Neural mechanisms for voice recognition [J]. 2010, 52(4): 1528-40.

[3] KAUR P, KRISHAN K, SHARMA S K, et al. Facial-recognition algorithms: A literature review [J]. 2020, 60(2): 131-9.

[4] BARRON U G, CORKERY G, BARRY B, et al. Assessment of retinal recognition technology as a biometric method for sheep identification [J]. 2008, 60(2): 156-66.

[5] ROHS R, WEST S M, SOSINSKY A, et al. The role of DNA shape in protein–DNA recognition [J]. 2009, 461(7268): 1248-53.

[6] THECKEDATH D, SEDAMKAR R J S C S. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks [J]. 2020, 1(2): 79.

[7] LEARNED-MILLER E, HUANG G B, ROYCHOWDHURY A, et al. Labeled faces in the wild: A survey [J]. 2016: 189-248.

[8] DING X, ZHANG X, MA N, et al. Repvgg: Making vgg-style convnets great again; proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, F, 2021 [C].

[9] AL-QIZWINI M, BARJASTEH I, AL-QASSAB H, et al. Deep learning algorithm for autonomous driving using googlenet; proceedings of the 2017 IEEE intelligent vehicles symposium (IV), F, 2017 [C]. IEEE.

[10] WANG Q, WU B, ZHU P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks; proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, F, 2020 [C].