Research and Application of Facial recognition Algorithm in Audit Investigation

Hairong Wang¹,², Weibo Li¹,²,*, Quan Wan¹,², Hua Yan¹,², Rui Xiang¹,²

¹School of Computer Science & Engineering, Wuhan Institute of Technology, Wuhan, China
²Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China

*Corresponding author: liweibo@wit.edu.cn

Abstract. Aiming at the application scene of face image recognition in audit investigation, firstly, a face data set with corresponding environmental interference is simulated to increase the diversity of data; in the preprocessing stage, the improved method of adaptive histogram illumination balance and simulated glasses covering is used for image enhancement; in the model training stage, an optimal weight reoveraging model training algorithm is proposed. The experimental results show that the accuracy, robustness and efficiency of face recognition in the application scene are improved by the improvement of image enhancement preprocessing and the model training of optimal weight reoveraging.

Keywords: deep learning, face recognition, audit investigation, model training.

1. Introduction

Traditional facial recognition methods include facial recognition based on geometric features, template matching, subspace and statistics. Now the mainstream facial recognition method is based on the deep convolution neural network method, although in the detection speed and occupied performance resources have no advantages of traditional methods, but in the detection accuracy, it has obvious advantages and better application prospects [2]. However, there are still many problems to be solved in the practical application of facial recognition.

In the meeting Compliance Investigation of flight inspection, some scenes involving personnel audit, such as the identification of key persons (lecturers and other important participants) in the meeting scene. Auditors will judge the compliance and risk degree of the meeting according to the presence of key persons. As one of the most important links, on-site meeting personnel audit often needs to be accurate and real-time. But the general face image recognition algorithm in these complex and diverse scenes, due to uneven light, detection object face cover and other factors, it is easy to lead to detection failure, recognition error and other problems [3 – 5]. To solve these problems, literatures [6 – 9] use random masking method to improve the robustness of face masking, but it is not targeted for face. Literature [10] introduces that Retinex algorithm can be used to alleviate the problem of illumination, but it is not applied in facial recognition preprocessing.
Based on the application scenario of audit survey, in the preprocessing stage, histogram equalization and Retinex illumination equalization are proposed to alleviate the problem of uneven face illumination. The improved algorithm of simulated glasses covering is used to improve the accuracy and robustness of facial recognition with glasses. In the model training stage, according to the phenomenon that it often takes a lot of training time to obtain a satisfactory model effect in the process of convolutional neural network model training, this paper proposes an optimal weight overload model training algorithm to speed up the convergence rate of the model and improve the accuracy of the final model forming. The results of comparative experiments show that these improvements can make the model achieve better results, and have very good universality and application value.

2. Research on relevant theories and methods

2.1. Image enhancement processing

The ways of the image enhancement can make full use of the image information of the original data set. There are generally two ways to process image enhancement. The first is offline enhancement, i.e. dataset enrichment. And the second is online enhancement, which is pre-processing when the dataset is loaded.

In the process of deep learning image field model training, image enhancement is generally achieved in the pre-processing stage. Because the average dataset uses a lot of memory, it saves a lot of space. At the same time, it is also easy to use random processing, so that the trained model generalization performance will be better.

The significance of using local masking and histogram equilibrium and Retinex equilibrium in the pre-processing of model training lies in making full use of the information of the original image to realize the robustness of the model to local features and uneven lighting.

2.1.1. Partial masking. The difference between local cover and random cover is that local cover is more targeted and can improve the robustness of the model to specific local features. In facial recognition, in order to improve the robustness of the model to the glasses, we can use the partial masking interference algorithm of the glasses.

The glasses masking algorithm simulates an image of a person wearing glasses. In the process of implementing spectacle masking, the face detection algorithm of MTCNN (Multi-task convolutional neural networks, multitask convolutional neural network) first locates the coordinates of the two eyes of the face, and then aligns the faces according to that coordinate and adds the glasses to cover. In general, the two eyes are not at the same level. Based on the coordinates(x1, y1, and x2, y2) of the two eyes, the angle between them and the horizontal line can be calculated.

\[ \theta = \arctan \frac{y_2 - y_1}{x_2 - x_1} \]  

(1)

![Figure 1. Glasses interference effect](image)

Then according to the angle of the angle of the correction, you can achieve the effect of face alignment.

2.1.2. Histogram equilibrium. Histogram equilibrium enhances contrast by reassigning the grayscaness histogram of the original image to a histogram with uniform grayscaness through a change function.
The probability of each grayscale \( k \) appearing in the histogram in each mobile template can be defined as:

\[
P(r_k) = \frac{n_k}{N}, \quad k = 0, 1, \ldots, L - 1
\]  

(2)

Where \( N \) represents the total number of pixels in the image, \( L \) represents the grayscale of the image, represents the \( k \)-grade grayscale, and \( r_k \) represents the \( k \)-grade grayscale.

Then equilibrium is performed according to the gray-scale probability:

\[
S_k = T(r_k) = (L - 1) \sum_{j=0}^{L} P(r_j) = (L - 1) \sum_{j=0}^{L} \frac{n_j}{N}
\]  

(3)

The contrast effect is shown in Figure 2.

![Figure 2. Histogram equalization](image)

2.1.3. Retinex light balance. Retinex theory suggests that an object's image \( S \) is reflected by the object's surface reflecting the incoming light \( L \), while the reflectivity \( R \) is determined by the object itself and is not affected by the incoming light \( L \). The illumination of the image depends on the light source irradiated on the object, and the reflectivity depends on the object itself. According to Retinex theory, from a mathematical point of view, the image can be divided by the reflectivity to calculate the light. The image formation model based on the Retinex method is as follows:

\[
I(m, n) = R(m, n) \cdot L(m, n)
\]  

(4)

Where \( I(m, n) \) represents the image, \( I \) is taken between 0 and 255, \( R(m, n) \) is the reflectivity of the object, the reflectivity is between 0 and 1, \( L(m, n) \) is the light, the light is also between 0 and 255.

Eventually, you can sort out:

\[
R = \exp(\log I - \log L)
\]  

(5)

As can be seen from type (5), to estimate the reflectivity, you must estimate the degree against the image. As a result, a variety of filters can be used to estimate the degree of light. Filters allow you to smooth the image, and in most Retinex-based image enhancement methods, a smooth image can act as a light. This is the principle of single-scale Retinex algorithm (SSR).

In this paper, the Retinex algorithm is introduced in the image pre-processing stage to alleviate the problem of unealed lighting of face images. The image pre-processed contrast effect by Retinex is shown in Figure 3 (the original image on the left and the Retinex-processed diagram on the right).
2.2. Model optimization algorithm

A suitable optimization algorithm is like choosing a potential shortcut to quickly find the optimal solution to the model parameters when the model is trained. The significance of studying model optimization algorithm is to find such a shortcut, shorten the training time of model and save the resources of model training. The commonly used model optimization algorithm methods have random gradient drop SGD and Adam, which also have advantages in the process of model training. The recently released Radam also works very well.

2.2.1. Random gradient drop method SGD

The random gradient decline, which originated from the random approximation proposed by Robbins and Monro in 1951 and was originally applied to pattern recognition and neural networks, has become a mainstream and very effective method for solving large-scale machine learning optimization problems. It is mainly used to solve optimization problems similar to the following form of seism:

$$f(w) = \sum_{i=1}^{n} f_i(w, x_i, y_i)$$  \hspace{1cm} (6)

Where $f(w)$ represents the loss of the entire dataset, $f_i$ represents the loss of the first sample, $w$ represents the weight of the current model, $x_i$ represents the characteristics of the input i sample, and $y_i$ represents the label of the first sample.

Gradient drop method:

$$w_{t+1} = w_t - \eta_{t+1} f(w) = w_t - \eta_{t+1} \sum_{i=1}^{n} \nabla f_i(w_t, x_i, y_i)$$  \hspace{1cm} (7)

When $n$ is large, it can be time-consuming to calculate all the $\nabla f_i(w_t)$ of the iteration. The idea of a random gradient drop is $\nabla f_i$ to randomly select a calculation of the $\nabla f_{i_k}$ in each of the .

$$w_{t+1} = w_t - \eta_{t+1} \nabla f_{i_k}(w_t, x_{i_k}, y_{i_k})$$  \hspace{1cm} (8)

$$i_k \in \{1,2,3,...,n\}$$

The algorithm is in the desired sense of convergence when the algorithm is selected for the length of the $\eta_t = O\left(\frac{1}{t}\right)$ as a result of the $\mathbb{E}[\nabla f_{i_k}(w_t, x_{i_k}, y_{i_k})] = \nabla f(w_t)$

Notice that when $w_t$ is close to the very small value point $w_*$, the $\nabla f_i(w_t) \neq 0$, which results in a low accuracy of the random gradient drop method. Because of the variance, in order for the algorithm to converge, it is necessary $\eta_t$ gradually decrease with $t$. As a result, the function has a convergence speed of only $O(1/t)$ even under strong and smooth conditions.

SGD can update parameters with only one training batch of data per iteration, making the cost function smaller, with the advantage that the training is fast, but the convergence is slow.
2.2.2. Adaptive learning rate optimization algorithm Adam. Adam integrates SGD's first-order momentum with RMSProp's second-order momentum, which adds inertia in the process of gradient drop, making the gradient direction faster on the dimension and slower on the dimension where the gradient direction changes, thus accelerating convergence and reducing the oscillation.

One step:

\[ m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * \nabla f(w_t) \]  \hspace{1cm} (9)

Second step:

\[ v_t = \beta * v_{t-1} + (1 - \beta_2) * \left( \nabla f(w_t) \right)^2 \]  \hspace{1cm} (10)

Bias correction: In the pre-training period, the gradient weight is relatively small, the value of the weight needs to be revised to 1.

\[ \hat{m}_t = \frac{m_t}{1 - \beta_1} \]  \hspace{1cm} (11)

\[ \hat{v}_t = \frac{v_t}{1 - \beta_2} \]  \hspace{1cm} (12)

\[ (w_{t+1})_i = (w_t)_i - \frac{\eta}{\sqrt{v_{t+\epsilon}}} * \hat{m}_t \]  \hspace{1cm} (13)

Adam combines the advantages of Adagrad's ability to handle sparse gradients with Rmsprop's ability to handle non-smooth targets for large data sets and high-dimensional spaces.

2.2.3. RAdam optimization algorithm. The optimizer, RAdam, proposed by Liyuan Liu for The Improvement of Adam, combines the advantages of Adam and SGD to ensure convergence speed and is not easy to tune into the local optimal solution. Radam points out that the variance in Adam in the early stages of training can be very large. Formula (11) corrects the direction of the update, so the variance of the number of updates to the Adam parameter is also large. At the beginning of the training (\(\rho_t < 4\)), the SGD of the drive is used to update the parameters. Use it later to warm up the learning rate. The amount of updates for each step of the final parameter is:

\[ \Delta \theta = \eta \cdot r_t \cdot Adam(t) + 1 - r_t \cdot SGDMomentum = \eta \cdot \left[ r_t \cdot \frac{1 - \beta_1^t}{1 - \beta_1} \cdot \frac{m_t}{\epsilon + \sqrt{v_t}} + \frac{1 - r_t}{1 - \beta_2} \cdot m_t \right] = \eta \cdot \frac{m_t}{1 - \beta_1} \]  \hspace{1cm} (14)

3. Theoretical application and improvement

3.1. Image enhancement application and improvement

Face image recognition in the face image recognition project of audit investigation is easily affected by such factors as uneven illumination and covering. Therefore, in order to increase the robustness of the model to illumination and masking, three enhancement operations have been added during the data enhancement phase: local mask interference, histogram equalization, and Retinex light balancing. The coordinates of key points need to be detected by MTCNN during local masking interference. Because MTCNN face detection is expensive, it is further optimized here. That is, when the original dataset was previously cropped by MTCNN, the identified face keys (including two eyes, nose tips, and
two mouth corners, together with coordinates of five points) are saved in the text file. Then look for the coordinates directly in the text file and return them when the eye coordinates of the current picture are required for the spectacle cover. This processing reduces the number of MTCNN usages and reduces the time consumption of pre-processing.

In addition, for the added local mask interference, histogram equilibrium and Retinex light equilibrium three enhanced operations, each given a probability value $p_i$ to determine whether to use the operation. In this way, the sample can be interfered with to varying degrees, so that the sample data information is fully utilized. The images obtained through such pre-processing are:

$$y = \sum_{i=1}^{n} p_i \times t_i(x)$$ (15)

Where $p_i$ represents the probability of the first pre-processing, $n$ represents the quantity of pre-processing, and $t_i$ represents the $i$ pre-processing, so that the picture trained after a round of pre-processing can make the model more robust.

The pseudo-code of this algorithm is shown in algorithm 1.

Algorithm 1. Pre-processing improvements
Input: img, trans, prop
Output: trans_img
1. for $i$ in range($n$):
2.     $p$ = random()
3.     if $p < prop[i]$ then
4.         img = trans[i](img)
5. return trans_img

3.2. Model training methods based on optimal power overload
There are many factors that influence the effect during the model training stage, such as the initialization of learning rate, the selection of optimizers, and the design of training methods. Choosing a good training method can improve the training efficiency, speed up the convergence of the model, and save the time of the convergence of the model.

Both SGD and Adam can solve the convergence problem in training to some extent, but both tend to look forward to the optimal solution, which can easily lead to the loss of local or even global optimal solutions that have been traversed. This is typically the case in the image: the accuracy of the validation set will be maximized at some point during training, and the accuracy of the continuing training validation set will decrease. As shown in Figure 4.

![Figure 4. The effect of overfitting in training](image)
When this happens, the Epochs (the number of trainings for the entire dataset) setting is no longer effective because the validation set has converged. It is common practice to stop training and recalculate super parameters in the face of this problem. This approach also requires manual tuning, which is cumbersome and uncontrollable. The model training method based on optimal weighting is to save the weight of the highest accuracy or minimum loss (the weight corresponding to best in the figure above) during the training of SGD or Adam optimizer. Pre-set a smaller epochs to 50, and then, each time the automatic retraining process, load the weight of the last saved best model, repeated several times, to speed up the effect of model convergence.

\[
L(w_{best}) = \min_{i=0}^{\text{reload_epochs}} L(w_i)
\]  

(16)

Where \( w_i \) represents the loss of the current batch, \( L(w_{best}) \) represents the minimum loss value corresponding to the weight with the smallest loss. The algorithm pseudo-code algorithm 2 for this process shows:

**Algorithm 2: Optimal weight heavy load algorithm**

**Input:** Epochs, train_times

**Output:** model_file

1. init model
2. set pre_acc=0
3. for train_iter in range(train_times)
   4. if exist(model_file) then reload(model)
   5. init optimizer, scheduler, loss_fn,
   6. for epoch in range(epochs)
      7. Train(model)
      8. acc = Valid(model)
      9. if acc > pre_acc then
         10. model_file=save(model)
   11. return model_file

The algorithm executes the process in Figure 5.

*Figure 5.* Flow chart of optimal weight reload algorithm
4. Compare experiments

4.1. Preparation for the experiment
Experimental environment: graphics GTX 1660Ti, memory 6G, system windows 64, programming tool pycharm, language version python 3.6, deep learning framework pytorch 1.3, using drawing tool tensorboard.

Data set processing: After the data set is disrupted, take 10% of it as a validation set fixed, 90% as a training set for repeated training.

4.2. Pre-processing comparison experiments
Experimental settings: The experimental parameters in the pre-processing comparison experiment are: learning rate Lr s 0.001; data set LFW; Batch s 100; model InceptionResnetV1; optimizer Adam; scheduler MultiStepLR; LOSS: CrossEntropyLoss; pretrained: vggface2; model size: 91M.

The results of the preprocessed comparison experiment are shown in Table 1.

| Pre-processing          | Accuracy | Increment | Cumulative increment |
|-------------------------|----------|-----------|----------------------|
| base                    | 73.2     | 0         | 0                    |
| + Flip                  | 75.5     | 2.3       | 2.3                  |
| + Glasses cover         | 76.1     | 0.6       | 2.9                  |
| + Histogram Processing  | 77.2     | 1.1       | 4                    |
| + Retinex light balance | 78.9     | 1.7       | 5.7                  |

Experimental results analysis: From the experiment, we can see that the accuracy of left and right flip increased by 2.3%, the spectacle cover increased by 0.6%, a cumulative increase of 2.9%, histogram processing increased by 1.1%, a cumulative increase of 4.0%, and the Retinex equilibrium treatment increased by about 1.7%, a cumulative increase of 5.7%. The best-results training curve is shown in Figure 6. Experiments show that the data enhancement can significantly improve the robustness and accuracy of the whole model.

4.3. Training method comparison experiment
The data set used in the experiment is the face data set that LFW has been cropped to 112 x 112 by MTCNN face detection alignment, and the sampling method during the training is to sample sample elements from a given index list without playback, i.e. randomly sample the data from the original data set, to generate any combination of the undersails, so as to extract the data in data set by using the undersize.
In order to fully verify the validity and superiority of the optimal weight loading algorithm, this part of the experiment did three sets of experiments on the mobileNetV2, resnet18, shufflenet_v2_x2_0 models, each set of experiments for different optimizers (SGD, Adam, RAdam) and the two algorithms of weight training and general training. The parameters are set to epochs $s 300$, batch_size $s 30$, weight_decay $s 5e-4$, and the learning rate is $1e-2$. The number of weight overloads in the overloaded curve is set to 5. The results of the experiment are shown in Figure 8-10, wherein the left half of each graph is the curve of the weight-loaded experiment and the right half is the curve of normal training. The experimental data in Figure 8 are analyzed below.

In the optimizer for SGD training comparison process, after 300 epochs, the accuracy of the training set using overloaded algorithms has been completely reduced to 100%, the loss is close to 0, the fitting effect is very good. The accuracy above the verification set is already as high as 50% and the loss is controlled at 2.3. And the use of general training process training, the accuracy of the training set is still about 40% shock, loss is also about 2.5 shock, completely unfit, verification set accuracy is only about 30%, the loss is still 3.5. In contrast, under the same SGD optimizer and training conditions at the same interval, the accuracy of using heavy loads was 20% higher than that of verification sets without overloading, the loss was reduced by 1.2, the training set convergence speed was increased by 60%, and the loss was reduced by 2.5. The experiment obviously shows that the overload algorithm has more powerful convergence speed and convergence efficiency than not using overload algorithm.

In the optimizer for Adam training comparison process, after 300 epochs training, the use of heavy-duty training sets reached 83%, there is room to continue to rise, losses are slowly declining; The use of non-heavy training sets has been fitted to 100%, the loss has been reduced to 0; In contrast, in the absence of a training set fitted, the overloaded validation set is 0.28 higher than the non-overloaded validation set, not to mention the upward momentum behind the overload. It is proved that the combination of overload algorithm and Adam optimizer can also significantly improve the convergence speed of model training, inhibit the effect of overfit, and improve the accuracy of model verification set.

During the comparison of the optimizer training for RAdam, after 300 epochs training, the accuracy of both heavy and non-overloaded has converged to 100% and the loss has converged to around 0. On the verification set, the accuracy of the algorithm using overload is 72%, the loss is stable at about 1.8, the accuracy of non-overload is only 0.61%, but after the loss is reduced to a minimum value of 1.8, it continues to rise, stable at about 3.3, at this time the trained model has lost its generalization ability. In contrast, with both training sets converging, the overload algorithm is 0.11 more accurate and the loss is stable at 1.8, while the non-overload algorithm trains to the back with less and worse results. This experiment proves that the overload algorithm can train models with better generalization ability and higher accuracy, and the trained model is more stable, so that the generalization ability is not worse because of the training time is too long. From the analysis of the above experimental results, it can be concluded that the overload algorithm has good advantages for MobileNetV2 to use different optimizers. From Figure 9, Figure 10 can also be seen, for other models have the same applicable effect.

The final model results (epochs=300) are collated from the training graph above, as shown in Table 2:
Table 2. The performance of various models with different optimizers in algorithms with or without reloading

| model               | optimizer | reload | Traditional | Increment |
|---------------------|-----------|--------|-------------|-----------|
| MobileNetV2         | SGD       | 0.5414 | 0.2728      | 0.2686    |
|                     | Adam      | 0.654  | 0.4166      | 0.2374    |
|                     | RAdam     | 0.7523 | 0.5976      | 0.1547    |
| resnet18            | SGD       | 0.5676 | 0.4011      | 0.1665    |
|                     | Adam      | 0.6861 | 0.6354      | 0.0507    |
|                     | RAdam     | 0.7802 | 0.7926      | -0.0124   |
| shufflenet_v2_x2_0  | SGD       | 0.3695 | 0.2242      | 0.1453    |
|                     | Adam      | 0.7335 | 0.5347      | 0.1988    |
|                     | RAdam     | 0.7578 | 0.7059      | 0.0519    |

From Table 2, it can be learned that the accuracy of the model can be improved by using overload algorithms with different models using different optimizers, with the highest being close to 20%.

In a word, experiments shown that the method of using the training model with the optimal right heavy load can speed up the convergence speed of the model, improve the accuracy of the model and the generalization ability of the model, and inhibit the overfit of the training dataset, which has also a universal effect on different models.

5. Conclusion
In view of the face image recognition application of the audit investigation application, three enhancement operations, local mask interference, histogram equalization and Retinex lighting equalization, are proposed in the pre-processing stage. Among them, the partial cover interference is mainly analyzed and experimented with the glasses cover as an example, which improves the accuracy of the model by 0.6%. It is shown that the cover of random glasses can effectively alleviate the problem of not recognizing the problem caused by wearing glasses. At the same time, the side shows that random local covering can improve the robustness of the local characteristics of the model. Histogram equilibrium and Retinex equilibrium improved the accuracy of the model by 1.1% and 1.7%, respectively, indicating that these two enhancement methods can alleviate the effect of uneven lighting on face image recognition. In the stage of training facial recognition model, a training method with optimal right load is proposed, and after several sets of experiments, it is shown that the training algorithm can improve the convergence speed and the generalization ability of the model.
| Optimizer | Accuracy | Loss |
|-----------|----------|------|
| SGD       | ![Graph](image1) | ![Graph](image2) |
| Adam      | ![Graph](image3) | ![Graph](image4) |
| RAdam     | ![Graph](image5) | ![Graph](image6) |

**Figure 7.** MobileNetV2 Model training comparison

| Optimizer | Accuracy | Loss |
|-----------|----------|------|
| SGD       | ![Graph](image7) | ![Graph](image8) |
| Adam      | ![Graph](image9) | ![Graph](image10) |
| RAdam     | ![Graph](image11) | ![Graph](image12) |

**Figure 8.** Resnet18 Model training comparison
| Optimizer | Accuracy | Loss |
|-----------|----------|------|
| SGD       | ![Accuracy SGD](image1) | ![Loss SGD](image2) |
| Adam      | ![Accuracy Adam](image3) | ![Loss Adam](image4) |
| RAdam     | ![Accuracy RAdam](image5) | ![Loss RAdam](image6) |

**Figure 9. Shufflenet_v2_x2_0 Model training comparison**

**Acknowledgments**

This work was financially supported by xxx fund.

**References**

[1] Zhu Xiuping, Wu Xueyi, Liu Wenfeng. Review and Prospect of face recognition[J]. Computer & Information Technology, 2008(04): 53–56.

[2] Chen Xitong, Lu Tao. Efficient Face Recognition Algorithm Using Global Deep Separable Convolutional and Residual Network[J]. Journal of Wuhan Institute of Technology, 2019, 41(03): 276–282.

[3] Wang Huixing, Huang Bo, Gao Yongbin, e.t. Review of Face Recognition Methods with Partial Occlusion[J]. Journal of Wuhan University(Natural Science Edition), 2020, 66(05): 451–461.

[4] Liang Lin, he Weiping, Lei Lei, e.t. Survey on enhancement methods for non-uniform illumination image[J]. Application Research of Computers, 2010, 27(05): 1625–1628.

[5] Yan Chunnan, Zhang Di, Hao Youfei, e.t. Face recognition method under variable illumination based on LC-KSVD dictionary learning[J]. Transducer and Microsystem Technologies, 2020, 39(11): 44–46.

[6] Yang Lanlan, Gao Mingyu, Wang Chenning, e.t. Research on facial expression recognition method based on data enhancement[J]. Computer products and circulation, 2020(11): 128–129.

[7] Wang Zhiwei, Fan Yihong, Zhao Jinwen, et al. The Development and Research of the Classroom Monitoring Personnel Identification System Based on Faster R-CNN[J]. Computer Knowledge and Technology, 2020, 16(17): 35–38.

[8] Zhang Hanyu, Wu Zhihao, Xu Yong, Chen Bin. Face forensics detection method based on enhanced convolutional neural networks[J/OL]. Computer Engineering and Applications:1-7[2020-11-17].http://kns.cnki.net/kcms/detail/11.2127.TP.20200701.1040.016.html.

[9] Gan LAN, Shen Hongfei, Wang Yao, Zhang Yuejin.Dataset enhancement method based on
improved deep convolutional generative adversarial networks[J/OL]. Journal of Computer Applications:1-11 [2020-11-17]. http://kns.cnki.net/kcms/detail/51.1307.TP.20201015.1715.017.html.

[10] Liang Lin, He Weiping, Lei Lei, et al. Survey on enhancement methods for non-uniform illumination image[J]. Application Research of Computers, 2010, 27(05): 1625–1628.

[11] Zhang K, Zhang Z, Li Z, et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks[J]. IEEE Signal Processing Letters, 2016, 23(10):1499-1503.

[12] Liu Mingzhu, Su Taoyu, Wang Yining. Research on Compressed Multiscale Retinex Image Enhancement Algorithms[J]. Journal of Harbin University of Science and Technology, 2020(05): 93–99.

[13] Zheng Meilin, Wang Shan Research on Image Enhancement Method Based on RETINEX[J]. Journal of Jujiang University(Natural Science Edition), 2020, 35(03): 69–73.

[14] Shi Jiarong, Wang Dan, Shang Fanhua, Zhang Heyu. Research Advances on Stochastic Gradient Descent Algorithms [J/OL]. Acta Automatica Sinica: 1-17 [2020-11-17]. https://doi.org/10.16383/j.aas.c190260.

[15] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization.International Conference on Learning Representations, pages 1–13, 2015.

[16] Liu L, Jiang H, He P, et al. On the Variance of the Adaptive Learning Rate and Beyond[J]. 2019.