ISSN 2616-5775 Vol. 6, Issue 10: 18-23, DOI: 10.25236/AJCIS.2023.061003

A Face Recognition Method Using ResNet34 and RetinaFace

Lingfeng Ma[1,a,*], Zuqiang Long[1,b]

[1]*School of Physics and Electronic Engineering, Hengyang Normal University, Hengyang, Hunan, China*
[a]*3151761257@qq.com*, [b]*dragon51@126.com*
**Corresponding author*

Abstract: *A new face recognition method is proposed by utilizing ResNet34 and RetinaFace, which is based on a lightweight framework for Python named Deepface. The new method is used to improve two shortcomings in the related literature: (1) the susceptibility of face recognition to interference, and (2) the quite limited number of faces detected in an image. First, the RetinaFace detector is used to replace the common detector to get more facial feature points and expand the area for detecting faces. Thus, the number of faces detected in the same image is increased. Then, ResNet34 model is applied to replace the default model in Deepface to improve the anti-interference in face recognition. Finally, experiments demonstrate that the new method is superior to the default one.*

Keywords: *Face recognition, ResNet34, RetinaFace, Detection number of faces, Anti-interference*

1. Introduction

In recent years, face recognition has made great advances by using deep convolutional neural networks (DCNNs). As a research hotspot, DCNNs have received increasing attention in the field of face recognition[1]. A gradient-enhanced softmax supervisor is introduced to recognize faces based on a DCNN[2]. This supervisor mitigates the vanishing gradient problem in the common softmax classifier. A deep softmax collaborative representation-based network is applied to solve multiple problems of face classification[3]. This network boosts the recognition performance of state-of-the-art deep learning networks. A novel double additive margin softmax loss function (DAM-Softmax) is adopted to recognize faces[4]. This function has clearer geometrical explanations and can obtain highly discriminative features for face recognition. However, if the output data to be processed reach a high order of magnitude, the above neural networks will generate the vanishing gradient problem. Meanwhile, as a face detector, the Haar cascade classifier of Opencv may result in a high-frequency false detection[5]. To solve the above problems in the default scheme, a new method of face recognition is proposed by means of Deepface. In the present study, ArcFace loss can process huge amounts of data without the vanishing gradient problem by building the ResNet34 model. This is an approach to detect more faces with a low-frequency false detection in the same image by utilizing RetinaFace.

2. Related work

Face recognition is divided into four steps: (1) face image acquisition, (2) image preprocessing, (3) image feature extraction, and (4) image feature classification in sequence. In this study, face images are acquired by IVCam software, which can provide high-quality real-time video by using a USB cable to connect a mobile phone to a computer. IVCam has the advantages of high definition, low latency, and self-defined resolution. Compared with the default computer camera, the high-definition camera of a mobile phone can get clearer face images by means of this software. The face detector is responsible for carrying out the process of image preprocessing, while the face recognition model deals with image feature extraction and image feature classification. The default scheme of Deepface and the new method in this study are shown in Table 1.

ISSN 2616-5775 Vol. 6, Issue 10: 18-23, DOI: 10.25236/AJCIS.2023.061003

Table 1: Methods of face recognition.

Configuration	Scheme	
	Default Scheme	New Scheme
Face Detector	Opencv(Haar)	RetinaFace
Face Recognition Model	VGGFace	ResNet34
Loss Function	softmax	ArcFace

2.1. Image preprocessing

Image preprocessing is divided into three steps: (1) face detection, (2) face alignment, and (3) face normalization. Face detection is applied to extract the area of the face and remove the rest in the image. Face alignment is an important step to further improve the accuracy of face recognition after face detection. The mainstream method of face alignment is taking human eyes as monitoring points. When two eyes in the face image are detected from a non-horizontal state to the opposite, the alignment can be considered as successful. The process of image preprocessing is shown in Figure 1.

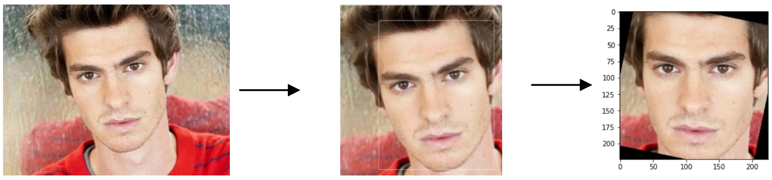

Figure 1: Process of image processing.

Face normalization is applied to solve the problem of illumination conditions affecting the accuracy of face recognition. The common method of face normalization is adaptive lighting system. A new method is presented by means of adaptive illumination normalization for face recognition [6]. What sets RetinaFace apart is that it has own image preprocessing method that integrates face detection, face alignment and face normalization.

2.2. Image feature processing

Image feature extraction is introducted to reduce the dimensionality of a massive image. It can reduce the difficulty of processing image data information. The mainstream method to carry out image feature classification is using a convolutional neural network. In the convolutional neural network, the input layer is responsible for receiving image data. The convolutional layer is in charge of the preliminary extraction of image features, the pooling layer carries out main extraction and the fully connected layer summarizes the extracted features for feature classification.

3. Plans of improvement

In the default scheme, the VGGFace model constructed by softmax loss may generate the gradient vanishing problem in a large-scale test. The problem has been researched by many scholars. An effective way is introducted in [7], and further studies can be found in [8-9]. To some extent, these ways can mitigate the gradient vanishing problem. However, the above methods are still not suitable for a large-scale face recognition test.

Haar is the common cascade for face detection. It may result in high-frequency false detection and low resistance to various interference factors. In the last few years, some scholars have researched this problem. A detection architecture and a non-human exclusion algorithm based on this cascade were proposed in [10] and [11], respectively. Although the performance of this cascade can be improved by many methods, its detection results are still not ideal in a large-scale test.

ISSN 2616-5775 Vol. 6, Issue 10: 18-23, DOI: 10.25236/AJCIS.2023.061003

4. Problems in the default scheme

4.1. Default scheme

As shown in Table 1, Opencv is an open-source library about computer vision in the default scheme. This library provides many functions that can be invoked quickly. Scholars can easily realize computer vision by using these functions. The Haar algorithm is a common face detection algorithm that is divided into four parts: (1) Haar-like feature, (2) integral graph method, (3) AdaBoost algorithm, and (4) cascade.

VGGFace model is a DCNN model invented by the Facebook team for face recognition research. This model has a structure with 22 layers and 37 depth units [12]. The structure of the model is shown in Figure 2.

Figure 2: Structure of the VGGFace model.

In the figure, image is the image data of the input layer, conv is the convolutional layer, maxpool is the pooling layer, and fc is the fully connected layer. The face similarity is measured by the method of cosine in the default scheme. The results of face recognition are shown in Figure 3.

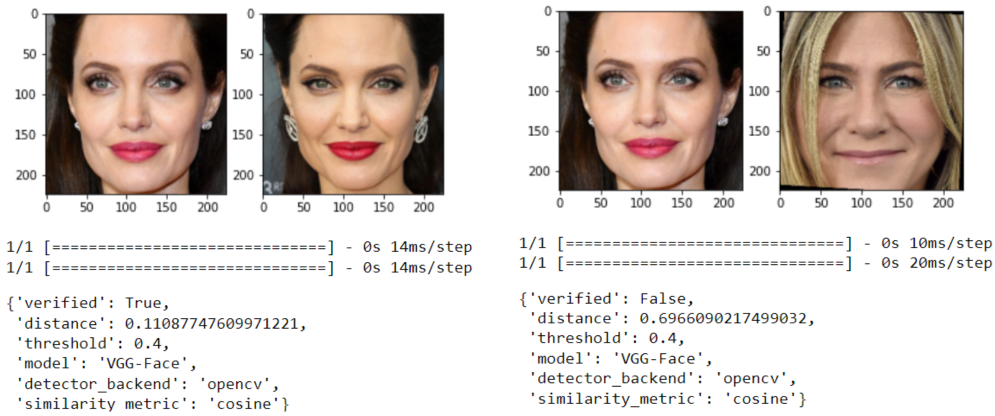

Figure 3: Results of face recognition.

In the above figure, True represents that the two faces are from the same person after recognition, and False means the opposite result. Distance is the diversity factor between two faces. If distance is less than 0.4, then the result of face recognition will be displayed as True. If it is not, it will be displayed as False.

4.2. Default scheme

4.2.1. Detector improvement

RetinaFace detector is adopted to solve the problem of the default one being easy to mislead when detecting faces. RetinaFace is based on the field face localization study, which is introduced in[13]. It can achieve pixel-level location of faces in various scenarios through combined external supervision and self-supervised multi-task learning. Compared with the default detector, the new one adds facial feature point location and adopts muli-task learning. Even in the case of many faces among one image, the new detector is still able to locate each face accurately.

4.2.2. Loss function improvement

ArcFace loss is proposed to handle the gradient vanishing problem. This function is based on softmax loss. The widely used formula of softmax loss is as follows:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{1}$$

Where, N is the number of samples, n is the number of categories, x_i is the depth feature of the ith sample, and y_i is the ith in the category. By convention, the embedded feature is set to $d = 512$, x_i belongs to R^d, and x_i is in class y_i. w_j represents column j of the weight of the fully connected layer of the neural network and b_j is the bias term. For ease of representation, b_j is set to 0 and the angle between x_i and w_j is set to θ_j. The value of $cos\,\theta_j$ is named as Logit, which can be converted to

$$cos\,\theta_j = arccos\left(\frac{W_j^T}{\|W_j\|}\cdot\frac{x_i}{\|x_i\|}\right) \tag{2}$$

Then, assume the L_2 norm is used to fix feature $\|x_i\|$ and weight $\|W_j\|$, and $\|W_j\| = 1$. The Logit is re-scaled onto the s hypersphere so that the feature normalization and weight normalization are only related to θ_j. By setting $b_j = 0$ and Formula(2) into Formula(1), and scaling on the s hypersphere, the loss function can then be rewritten as Formula(3):

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s\,\cos\theta_{y_i}}}{e^{s\,\cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{n} e^{s\,\cos\theta_j}} \tag{3}$$

The embedded features are distributed around the center of gravity of each feature on the s hypersphere. ArcFace adds an additive angular margin m between x_i and W_{y_i} to obtain $\theta_{y_i} + m$, which can obtain both intra-class compactness and inter-class diversity[14]. Putting $\theta_{y_i} + m$ in Formula (3) instead of θ_{y_i},the loss function obtained from above steps is ArcFace loss. This function can be expressed as

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s\left(cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(cos\left(\theta_{y_i}+m\right)\right)} + \sum_{j=1, j\neq y_i}^{n} e^{s\,\cos\theta_j}} \tag{4}$$

To test the effects of the default loss function and the new one, six people with different faces are selected to extract the data of face. These people are divided into six categories and each category includes 1200 samples. Two-dimensional features of the default loss function and the new one are respectively trained to the embedded network. The results of training are shown in Figure 4.

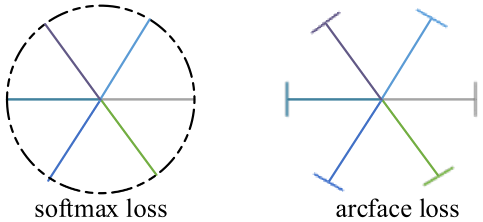

Figure 4: Results of training.

As shown in Figure 4, softmax loss can classify different faces to some extent. However, there are obvious overlaps at the boundaries of different categories and the distance between different classes is too short. The faces cannot be accurately separated into six categories. On the contrary, ArcFace loss can accurately distinguish six categories of different faces. The distance between different categories increases significantly and there is no overlap between the boundaries of classes.

5. Results and analysis

The testing platform is Jupyter Notebook, which is a Python interpreter that is widely used in deep learning. This interpreter runs directly on the web side and can run python programs step by step. The face image database named LFW is used to test two different schemes. The new scheme is reproduced in a third-party library of Python named keras for convenience. Figure 5 shows the results of recognition

ISSN 2616-5775 Vol. 6, Issue 10: 18-23, DOI: 10.25236/AJCIS.2023.061003

accuracies after testing by means of the two schemes. The results of large-scale face detection are shown in Figure 6 and 7.

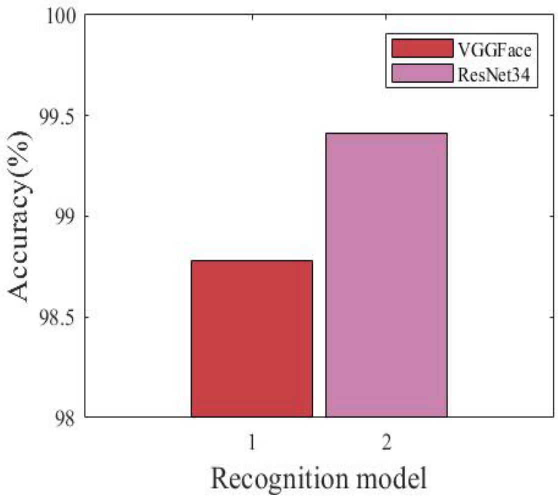

Figure 5: Results of recognition accuracies.

As shown in Figure 5, the recognition accuracy of the default scheme is 98.78% in LFW and the new one reproduced in keras is 99.41%. The face recognition accuracy of the new scheme is 0.63% higher than the default one. The performance of the ResNet34 model is better than the VGGFace model for face recognition by using ArcFace loss.

Figure 6: Results of large-scale face detection with Opencv.

Figure 7: Results of large-scale face detection with RetinaFace.

As shown in Figure 6 and 7, the detection number of faces in the image by the default scheme is no

ISSN 2616-5775 Vol. 6, Issue 10: 18-23, DOI: 10.25236/AJCIS.2023.061003

more than 20. On the other side, 50 faces can be detected simultaneously in the same image by using RetinaFace. In conclusion, the new scheme not only improves the accuracy of face recognition but also greatly increases the number of faces detected in the same image. The new scheme is more suitable for the complex face detection environment in reality and provides more effective information for human computer interaction. The experiments show that the new scheme performs better than the default one.

6. Conclusions

Opencv has various advantages for face detection in common environments. However, it may generate high-frequency false detection or low detection accuracy in multi-person environments. Meanwhile, the VGGFace model built by softmax loss may result in the vanishing gradient problem when processing massive data. To solve the above problems, RetinaFace detector has been introduced to replace Opencv, and ArcFace is utilized to replace softmax as the loss function. RetinaFace detector can detect more faces in the same image. ArcFace loss can reduce the frequency of false detection and improve the accuracy of face detection. The experimental data shows that the number of faces detected in the same image are improved from 20 to 50 by means of the new method. The accuracy of face recognition is improved from 98.78% to 99.41%.

Acknowledgements

This work is supported by Natural Science Foundation of Hunan Province (Project No. 2022JJ30104) and the Key project of Hunan Provincial Department of Education (Project No. 21A0439).

References

[1] Grm, Klemen, et al. "Strengths and weaknesses of deep learning models for face recognition against image degradations." Iet Biometrics 7.1 (2018): 81-89.

[2] Sun, Linjun, et al. "Gradient-enhanced softmax for face recognition." IEICE TRANSACTIONS on Information and Systems 103.5 (2020): 1185-1189.

[3] Vo, Duc My, Duc Manh Nguyen, and Sang-Woong Lee. "Deep softmax collaborative representation for robust degraded face recognition." Engineering Applications of Artificial Intelligence 97 (2021): 104052.

[4] Zhou, Shengwei, et al. "Double additive margin softmax loss for face recognition." Applied Sciences 10.1 (2019): 60.

[5] Serengil S I, Ozpinar A. Lightface: A hybrid deep face recognition framework[C]//2020 innovations in intelligent systems and applications conference (ASYU). IEEE, 2020: 1-5.

[6] Vishwakarma V P, Dalal S. A novel non-linear modifier for adaptive illumination normalization for robust face recognition [J]. Multimedia Tools and Applications, 2020, 79(17): 11503-11529.

[7] Zhu Q, He Z, Zhang T, et al. Improving classification performance of softmax loss function based on scalable batch-normalization [J]. Applied Sciences, 2020, 10(8): 2950.

[8] Wang C, Song L, Wang G, et al. Multi-scale multi-patch person re-identification with exclusivity regularized softmax[J]. Neurocomputing, 2020, 382: 64-70.

[9] Li X, Wang W. Learning discriminative features via weights-biased softmax loss [J]. Pattern Recognition, 2020, 107: 107405.

[10] Hyun J, Kim J, Choi C H, et al. Hardware architecture of a Haar classifier based face detection system using a skip scheme [C]//2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2021: 1-4.

[11] Deng X, Zhu Z, Chang J, et al. Algorithm research of face recognition system based on haar[M]//Advances in Computer Science and Ubiquitous Computing. Springer, Singapore, 2021: 317-323.

[12] Parkhi O, Vedaldi A, Zisserman A. Deep face recognition[C]//BMVC 2015-Proceedings of the British Machine Vision Conference 2015. British Machine Vision Association, 2015.

[13] Deng J, Guo J, Ververas E, et al. Retinaface: Single-shot multi-level face localisation in the wild[C] //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 5203-5212.

[14] Deng J, Guo J, Xue N, et al. Arcface: Additive angular margin loss for deep face recognition[C] //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4690-4699.