A Face Recognition Algorithm For Edge Detection Equipment

Shengan Zhou[1,*], Zhiming Huang[2]

[1]Guangdong vocational Institute of Public Administration, Guangzhou 510800, China;

[2]Guangzhou qiusuo Electronics Co., Ltd, Guangzhou 510430, China

Abstract

In order to overcome the problems of missing detection, false detection and poor recognition effect of edge intelligent devices in face detection, a new face recognition method based on ETLB is proposed. This method can achieve face recognition and feature calibration, call a large number of trained face model interfaces, and have good robustness to occlusion. By testing the process of face detection, feature point calibration, feature vector extraction and comparison, the experimental results show that this method is superior to traditional image recognition methods, and can effectively improve the detection sensitivity, recognition accuracy and recognition effect. It can effectively solve the problem of poor real-time performance in dynamic image recognition.

Keywords

Face detection; feature point calibration; feature vectors; face recognition.

1. Introduction

Using the analysis and comparison results of face visual feature information to carry out face recognition is an important computer technology [1], an application model of computer vision image recognition technology, and also one of the hotspots of artificial intelligence in recent years. The use of edge intelligent devices can greatly reduce manual operations. At present, it can be applied to finance, security, stations, unmanned shops, public security systems and other fields. It is a biometric recognition technology based on face features, which is used for fast recognition and can be used to determine or locate identity. This technology can avoid some defects of traditional identity authentication, such as certificate, magnetic card, IC card and password.

This paper discusses in detail the deep learning and face recognition algorithm based on the improved convolutional neural network, which we call eltb algorithm. In this method, face detection is firstly carried out to locate and calibrate feature points, and then feature points and known faces are recognized as deep learning training parameters. Compared with the traditional recognition methods, this method can realize face detection and recognition, and overcome the problems of recognition errors and low recognition accuracy. The experimental results show that this method has superior performance.

2. Etlb Face Recognition Algorithm

2.1 Process

Face recognition includes face detection, feature poi(2) face recognition algorithm and deep learning.

Face recognition includes face detection, feature point extraction, face calibration, face comparison, etc. it can detect, track and analyze the face of each frame of the face image or static image in the video stream, and convolute and merge the images according to the principle of convolutional neural network dichotomy. The softmax activation function is used to determine whether there is face classification. We can detect whether the image contains one or more human faces, and calibrate the detected faces according to the neural network regression algorithm. Gray the image, extract face features, select feature points, and calibrate the face through affine transformation. After feature digitization, according to the similarity calculation method in neural network, the calibrated face is compared with the face in the training database. Finally, we can estimate whether the tested person

can be identified and whether his personal information can be found. The algorithm flow chart is shown in Fig. 1..

Figure1. Face recognition algorithm

2.2 Deep learning

Deep learning is a new modeling method in information analysis,it uses the modeling technique of establishing the relationship between input and output data, the multi-layer neural network framework is used to simulate the complex model. The deep learning method based on the convolutional neural network is a kind of calculation involving convolution or correlation calculation. Feedforward neural network algorithm with deep structure has been widely used in face feature extraction.[2] The principle is to clip the face image and input it to the input layer, then output a two-dimensional vector corresponding to the clipping matrix. Input data at the convolutional layer, the convolution kernel is used as a sliding window to slide over the image matrix, it can magnify similar features to the convolution kernel, the convolution kernel extracts the features of the tensor. Then pooling reduces the size of output feature vectors. After several convolution and pooling layers, the features of human face are finally output through the full connection layer to realize face comparison.[3]

3. Algorithm Implementation

3.1 Open source library

Opencv library is a cross platform computer vision library with a large number of Python interfaces, which can realize image processing. Opencv's face detection classifier is a cascade classifier based on Haar algorithm.

By creating histogram equalization of gray-scale images, haardectobjects function is used to detect the face calibration rectangular area of cascaded classifiers. For face recognition, we should collect data and train classifiers. [4] Through experiments, it is found that there are missed detection and false detection in the classifier file. Therefore, this study uses Dlib library for face detection and recognition, and calls the image processing function of OpenCV library for image processing.

Dlib library is a cross platform C + + open source library for machine learning, which contains many common machine learning algorithms and a large number of graphic model algorithms. [5] It supports a large number of numerical algorithms. We use the Dlib basic library, write the corresponding neural network structure, form our own etlb library, and then pass the pre trained parameters to the called neural network. It includes face detector, trained face key point detector and face recognition model, and provides programming interface for easy calling. We use Etlb deep learning face recognition technology, and the accuracy of outdoor face detection database benchmark is 97.28%.

3.2 Environment configuration

The use of Etlb library to realize face recognition in the early stage environment configuration is more complex, but the order of configuration, required software and various libraries are necessary. The installation of the Etlb library relies on CMake and BOOST. CMake is a cross-platform installation and compilation tool that can compile source code, make program libraries, and output a variety of makefiles or project files. Etlb and other C++ libraries need to be compiled into static libraries. To preinstall the compilation environment CMake, run CMake-gui.exe, configure the bin path of CMake to the system environment variable, and enter CMake -version in cmd to test that the installation is correct. Boost library is a cross-platform free and open source C++ library, it can improve the development efficiency of software.

Before installing Etlb, it must install wheel in Python3.7 directory under the Lib folder to facilitate Python module installation. It can use cd command to locate the current path to the Dlib directory, execute command:PIP Install dlib, then execute command: PIP List to see if it is installed successfully.

Etlb is used for face recognition, and there are many libraries to be imported in the program, such as Numpy, Scipy, OpenCV, Imageio and so on.To install libraries in Site-packages directory, then execute:PIP Install plus library name.

4. Experimental analysis

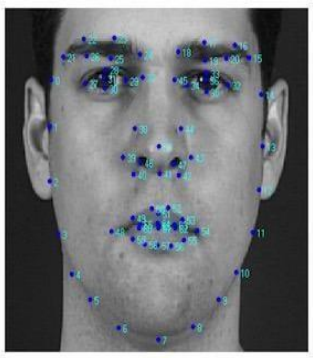
4.1 Face detection

Face detection is the preprocessing of face recognition. In order to detect the presence of face in complex scenes and accurately demarcate the position and area of face image, face detection must be carried out firstly.In this process, the first step is image input scanning, face detection is carried out from each frame image of static picture or dynamic video, and then face discrimination is implemented. If the face is found, the area where the face is located and is globally detected, and the positions of all faces are demarcated according to the input image information, the coordinates of face frame are output [6].

This paper uses the principle of face detection based on convolutional neural network. During face detection, it calls the face recognition library and the image processing library, uses the forward face detector dlib.get_frontal_face_detector()→to→implement. Face_detector () is based on HOG gradient histogram (direction), normalize color space, divide the detection window into cells of the same size unit, using gradient information to reflect the information of image target edge, combine linear classifier, image pyramid and sliding window detection scheme by the size of the local gradient of the image and appearance of the local shape characteristic.[7]

4.2 Face feature point location

In order to obtain more information about the detected face image, we need to locate feature point. Face feature point location is a technology of face image analysis. By locating, we can obtain facial contour boundary points, such as eyebrows, eyes, nose tip, mouth corner and other feature points, and accurately indicate face position. Etlb library is used to train the key point detector model of human face, Shape_predictor_68_face_landmarks.dat→is→used→to calibrate sixty-eight feature points and store coordinates of each point. We use OpenCV to process image, use cv2.circle () function to draw sixty-eight points and mark the number 0ne to sixty-eight with cv2.putText() function. Before traversing the fixed point coordinates of face, the images are read in and loaded into array form and then converted into grayscale images. The images are stored in reverse order BGR in OpenCV, which is different from the traditional RGB color channel used in the training set. Conversion is carried out by means of average method, so that the values of the three channels at the same pixel position are equal[8].We use Cv2.cvtcolor function to convert color space. Each static face image and each frame in the video can be marked with sixty-eight feature points[9], as shown in Fig. 2.



(a)Static facial feature points calibration and location

(b)Dynamic facial feature points calibration and location

Figure2. Static and dynamic face detection and feature location

4.3 Face alignment

There may be a variety of face pose in the picture, such as raising head,traightening face, siding face and so on. If you want to extract correct face feature, you need to do face alignment operation, straighten face, make it in the center of image, try to eliminate the error. The output of face features is more comparable, improve alignment precision and accuracy, reduce the pressure of calculation. Face normalization is usually carried out by using affine transformation matrix to select some feature points based on the calibrated sixty-eight feature points. This calibration method is helpful to improve the tilted face information, but the side face cannot be positive because it's short of the face information.[10] Before face normalization, we use Shape_predictor_68_face_landmarks.dat model to detect the distribution of sixty-eight key feature points of the face. According to the calibration information of sixty-eight feature points, we can locate five feature points, which are left eye, right eye, tip of nose, left mouth angle and right mouth angle.[11] On this basis, we can calculate the deviation angle and coordinates of the center point, the angle is corrected by affine transformation function, the algorithm is iterated until convergence. The image affine transformation is achieved through a series of atomic operations such as translation, scaling, flipping, rotation and shearing. It is a linear change from two-dimensional coordinates to two-dimensional coordinates, maintains the 'flatness' and 'parallelism'→of→the→image. MatgetRotationMatrix2D(Center, Angle, Scale) function is used to obtain the transformation matrix M, Center shows the center point of rotation, Angle shows the rotation Angle, Scale shows the scaling factor. The matrix formula is as follow:is one-dimensional 128 vector,M is

$$M = \begin{bmatrix} \alpha & -\beta & center.x\ (\alpha - 1) - center.y\ \beta \\ \beta & \alpha & center.x\ \beta + center.y\ (\beta - 1) \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

α=scale*cos(angle), β =scale*sin(angle), (center.x, center.y) represents the axis of rotation axis. The algorithm idea of affine transformation: firstly, the parameters are calculated. Then the affine matrix is calculated. Finally, the affine matrix is used as the parameters in warpAffine() function to carry out affine transformation.

4.4 Face feature vector extraction

The corrected faces are sent into the face classification and recognition network, and a layer of the classification network is extracted as the feature layer of the face to obtain the face features.128D feature vector is generated through quantization and calculation of features by convolutional neural network, each face is represented digitally. Each face image is made up of certain pixel, each pixel with RGB three channels, each pixel value range is 0 to 255 bytes, accumulative total matrix bytes as input data, generate an approximation function f * (x) in deep learning, the input data turns into vector can be used as a feature of multiple numerical classification characteristic value. [12]

In this paper, we use a model interface of deep Residual Network (ResNet) that has been trained by Dlib, it is dlib_face_recognition_resnet_model_v1.dat.It is a classification network for face feature extraction, which can be used to convert face images into 128D feature vectors and return value list via return_128d_features().In this way, similar faces are more similar and unsimilar faces are further away.[13]

The photos of a person have difference in different periods, the cut face image recognition rate may be reduced. In order to draw reliable characteristic vector, we need to calculate average of 128D characteristic vectors (such as Formula 2), the formation of the average face easy to identify, it can eliminate some photos what will be enlarged or reduced to face some features, save the results to CSV file.

$$\varphi = \frac{1}{M}\sum_{i=1}^{M} X_i \tag{2}$$

Φ Is one-dimensional128vector,M is the number of faces

4.5 Face matching recognition

Face recognition is to compare the similarity of the face features to be recognized with the face features in library, we can judge the identity information of the face. After 128D feature vector extraction, calculating Euclidean distance between face feature matrix and face feature matrix of trained database. When the distance< the specified threshold value, its feature similarity is high, it is considered to be the same person. When the distance>the specified threshold value, its feature similarity is low, it is considered not to be the same person. Through multiple iterations, if the result is not the same person all the time, it is considered that this person cannot be found in the database. Generally speaking, the greater the Euclidean distance, the lower the similarity between two eigenvalues, the less possibility they are considered to be the same person. When the identity comparison is 1:N, the larger N means the more feature vectors in the database and the lower the efficiency of face recognition. When recognizing face targets, feature comparison within the set is required. Euclidian distance is used to calculate the similarity between feature vectors.

The two vectors are set as x,y, n=128, the calculation formula of Euclidian distance is as follow[14]:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3}$$

Face_recognition.Face_distance() function is used as interface in the Dlib library to calculate Euclide distance.Face_recognition.compare_faces() function is face matching function, it is used to compare the test image with the picture encoding list in the test library. This function confirms whether it is the same face according to the threshold value, the default discrimination threshold is 0.6, excessively high threshold is easy to confuse face recognition. After repeated tests, the threshold value is modified as 0.4. Main algorithm idea: import face_recognition function, use FOR loop to load all images and images to be tested in the same test library folder, convert all loaded image pixels into facial coding, use loop algorithm to calculate Euclide distance and compare the similarity between test images and all images in the test library, and use logical value to express the result.

5. Conclusion

This paper proposes a face recognition method based on etlb. The algorithm is simple, and its advantage is that it uses an Open-Source Library as the recognition interface, which can achieve better face detection and recognition for still pictures, dynamic videos and real-time camera videos, with high accuracy and efficiency. In extreme cases, such as large deflection, similar color and environment, time-consuming problems need to be further studied to improve detection efficiency.

References

[1] Factors Affecting the Use of Facial-Recognition Payment: An Example of Chinese Consumers. [J] . Wen Kun Zhang,Min Jung Kang. IEEE Access . 2019.

[2] LI Yuelong,JIN Ya,et al. A survey of face feature point extraction methods[J]. Journal of Computer Science. 2016,39(7):1356-1374

[3] Daniel Sáez Trigueros, Li Meng. Face Recognition: From Traditional to Deep Learning Methods[J]. Computer Science.2018

[4] LIAO Chunping,CHEN Xuefang,et al. Face recognition system based on OpenCV[J]. Journal of Dongguan Institute of Technology. 2017,24(03):33-38

[5] LIANG Jing. Face recognition based on Python Library[J]. Network Security Technology and Application. 2019,(07):46-47

[6] GAO Zhuoyu. The basic principle and application of face recognition technology[J]. Practical Electronics. 2019(14):41-42

[7] [7]ZHANG Zhiling. Face recognition based on deep learning in Python[J]. Electronic Commerce. 2018,(05):47-47+96.

[8] WANG Xiaohong,HAN Jiao,LI Shanshan. Realtime video face detection and tracking based on face detector[J]. Information Communication. 2019, 2(194):56-57.

[9] MA Haodong,WANG Tianhao. Analysis of smile intensity based on two sets of eigenvalues of face 68 feature points [J]. Electronic Instrumentation Customers. 2018,25(01):12-16

[10] LI Haiyan,Xu Tingrong, et al. Multi pose face correction and recognition based on affine transformation[J]. Computer Application Research. 2014,31(04):1215-1219.

[11] HUANG Jiacai,KUANG Wenteng,MAO Kuancheng. Fatigue driving detection based on face key points[J]. Journal of Nanjing Institute of Engineering (NATURAL SCIENCE EDITION). 2017,15(4):8-13.

[12] Florian Schroff. FaceNet: A Unified Embedding for Face Recognition and Clustering[J]. Computer Science.2015

[13] XU Haohao.Face recognition based on Python deep learning[J]. Modern Information Technology. 2019,3(17):88-90.

[14] HUANG Wufei,Zeng Jichang.Face recognition based on laboratory security service robot[J]. Computer and Modernization. 2019,(07):61-64.