Design of Face Recognition System Based on Convolutional Neural Network

Kezhu Tao
School of Automation
Chongqing University of Posts and Telecommunications
Chongqing,China
379032359@qq.com

Yonglu He
School of Automation
Chongqing University of Posts and Telecommunications
Chongqing,China
575509448@qq.com

Caihong Chen
School of Automation
Chongqing University of Posts and Telecommunications
Chongqing,China
1195253802@qq.com

Abstract: This paper designs a service robot-oriented face recognition system. For mobile robots, face recognition is a very important function among them. The system includes three parts: face acquisition and preprocessing, model establishment, and model training. Among them, the face collection link uses the face detection function in opencv, and optimizes the interference factors to establish its own data set. A convolutional neural network was designed and constructed, and the model was trained on its own data set. The accuracy rate obtained on the test set was 97.63%. Finally, the trained model was applied to the actual system. The system model is simple, occupies a small amount of memory, and can be applied to the actual application scenario of the robot moving forward, which can quickly and accurately detect and recognize human faces.

Keywords—face detection, face recognition, convolutional neural network

I. INTRODUCTION

The face recognition research includes the following five aspects: (1) face detection. It is to find out the coordinates of the face and the area of the area occupied by the face from different situations. This method is affected by light intensity, image noise, head yaw, face size, mood, picture imaging equipment quality, and various occlusions [1]. (2) Face representation. This step is to extract the facial features of the person, determine the detected face and the existing face description in the database, including face geometric features (such as Euclidean distance, curvature, angle, etc.), algebra features (such as matrix feature vectors) Etc.), fixed feature templates, feature faces, moiré maps, etc. (3) Face recognition. The object to be tested is compared with the existing face image in the database, and the result is obtained. The key to this step is to select the appropriate face expression method and matching algorithm[2,3].

Scholars have conducted in-depth research on face recognition algorithms, and proposed many classical methods such as feature face [4,5], elastic map matching and LBP. For example, Li Qianyu constructed a feature extractor that can extract deep features automatically, and performed ZCA whitening on the data, and proposed a face recognition algorithm to improve the deep network[6]. The convolution kernel used in the algorithm mainly passed unsupervised learning. Obtained, the method has improved accuracy and performance compared to the conventional method. Zhang K. et al. [7] proposed a depth cascade task framework for various pose, illumination and occlusion problems in face detection, using a coarse to fine way to predict face and landmark positions and ensuring real-time performance. Yang et al. [8] studied an improved SSD convolutional network video object detection model[9,10], which can effectively identify targets in distant scenes in video, and reduce the amount of computation and reduce the memory resource consumption of hardware[11].

For mobile robots, it is necessary to make a quick and accurate response to interactive objects, so face recognition is a very important function. In this paper, a face recognition system is designed based on convolutional neural network. When the robot recognizes the identity of the visitor, it can make a corresponding greeting or action to them.

II. DESIGN OF SYSTEM

The system includes three sections: face acquisition and preprocessing, model establishment, and model training. The face collection module uses the face detection function in opencv, and has established its own training data set through preprocessing. The convolutional neural network was designed and constructed, and the model was trained on its own training data set, and the trained model was applied to the actual system.

A. Data Collection

The data set used in the training of this system is a self-built data set. There are 22 people in the self-built data set, 100 photos per person, and the images collected are images of different angles and different expressions of the same person.

The video stream captured by the camera is placed in the Mat container, and the captured video image is a color image. The BGR image is converted into a grayscale image and histogram equalization is performed to complete the preliminary processing. The processed image is classified using the haarcascades classifier. The haarcascades classifier is a cascade classifier that utilizes Haar features (also known as Haar-like features). The face is marked by Haar-like features, and the Haar-like eigenvalues are accelerated by using the integral graph. The AdaBoost algorithm is used in the training process to generate strong classifiers to distinguish between faces and non-human faces. After a series of strong classifiers are cascaded, the haarcascades classifier is obtained. Load the haarcascade_frontalface_alt.xml file in opencv. The detectMultiScale() function can detect the face with the classifier and input the video stream detected by the camera into Mat container. This is the input image to be detected. The output is Mat faces, which is the detected face. The target sequence, the image reduction ratio in this article is set to 1.1. In order to improve the accuracy of face detection, when multiple rectangle detection windows complete feature matching at the same time, it is recognized as a face. Here, a detection threshold is set, and this detection threshold also affects the detection effect. The rectangle function can draw a face detection frame on the video in the specified format. Finally, the collected data was stored in a personal database.

978-1-7281-4094-0/19/$31.00 ©2019 IEEE

This project collected 100 pictures of 22 people each as a self-built database.

B. Optimization of interference factors

For the picture blurring in the collected pictures, Wiener filtering and Laplacian edge detection were used to process the picture blurring caused by relative motion during the picture collection. At the same time, the collected non-positive face images are processed by affine transformation, which can correct a certain side face angle and prepare for subsequent face recognition. Finally, the optimized image is scaled by equal proportions, and the image data is labeled and loaded into memory in the form of a multi-dimensional array.

After filtering the data, we can also take some data enhancement methods. Data enhancement can increase the amount of data trained, improve the generalization ability of the model, increase the noise data, and improve the robustness of the model. Common methods include zooming, truncating, random angle rotation, brightness contrast adjustment, etc. Several methods can be selected for data enhancement.

C. Data Loading and Preprocessing

Regarding the use of data sets, we need to take part out of the training network to build the recognition model; the other part is used to validate the model. Before the model is trained, you need to complete the following four steps:

1) Cross-validation is a commonly used accuracy test method in machine learning, which can improve the reliability and stability of the model. Usually, most of the data is used for model training in the process of cross-validation. A small amount of data is used to verify the model after training, and the verification result is compared with the real value. This is repeated until the verification result is the same as the real value. According to the principle of cross-validation, the data set is divided into three parts: training set, verification set and test set; 70% is randomly selected as the training set and 30% is used for the verification set in the training. In the later test phase, 40% was randomly selected as the test set.

2) According to the requirements of the back-end system running by the keras library, the dimensional order of the image data is changed. The keras back-end system used in the system is tensorflow;

3) One-hot code conversion of the data labels of the training set, the verification set and the test set to make it vectorized

4) Finally, the data set is normalized, the purpose of which is to improve the network convergence speed, reduce the training time, and adapt the activation function between the values (0, 1) to increase the discrimination.

III. Model Building and Training

A. Model Building

The convolutional neural network (CNN) belongs to the feedforward neural network. It is generally formed by stacking convolutional layers, convergence layers and fully connected layers[12]. It is trained based on the back propagation algorithm and is mainly used to process image information. When training images using a fully connected feedforward neural network, there are problems with too many parameters and difficulty in extracting locally invariant features[13,14]. As shown in Fig.1, Convolutional neural networks have local connections, weight sharing and equivariance represent three structural features. Therefore, these characteristics make the convolutional neural network have a certain degree of translation, scaling and rotation invariance[15].

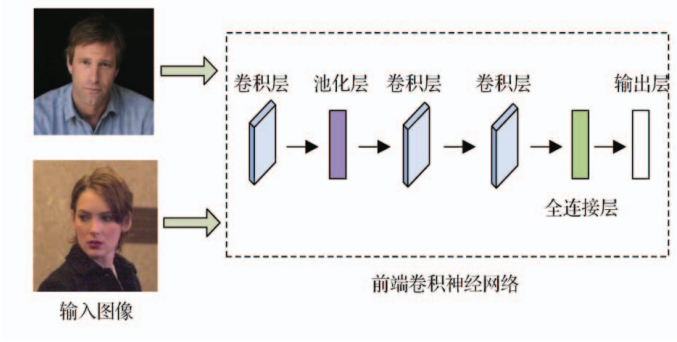

Fig. 1. Convolutional neural network

This network model consists of 14 layers, including 4 convolution layers, 4 activation function layers, 2 pooling layers, 2 Dropout layers, 1 fully connected layer, 1 Flatten layer, and finally connected 1 Classification layer, as shown in Table I.

TABLE I. Model Structure

Layer(type)	Output Shape	Param#
convolution2d_1(Convolution2D)	(None,64,64,32)	924
activation_1(Activation)	(None,64,64,32)	0
convolution2d_2(Convolution2D)	(None,62,62,32)	10759
activation_2(Activation)	(None,62,62,32)	0
maxpooling2d_1(MaxPooling2D)	(None,31,31,32)	0
dropout_1(Dropout)	(None,31,31,32)	0
convolution2d_3(Convolution2D)	(None,31,31,64)	21975
activation_3(Activation)	(None,31,31,64)	0
convolution2d_4(Convolution2D)	(None,29,29,64)	42308
activation_4(Activation)	(None,29,29,64)	0
maxpooling2d_2(MaxPooling2D)	(None,14,14,64)	0
dropout_2(Dropout)	(None,14,14,64)	0
flatten_1(Flatten)	(None,12544)	0

dense_1(Dense)	(None,22)	395194
Total params: 5,389,014		
Trainable params:5,389,014		
Non-trainable params: 0		

As shown in Table I, the constructed convolutional neural network can be divided into four segments. The first segment consists of a convolutional layer and an activation function layer. The input of the first convolutional layer is the input layer, not counting into the model, the size is 64*64*3, and the first convolutional layer has 32 volumes. The kernel, each convolution kernel has a size of 3*3, and then is output to an activation function layer. Since the relu function converges faster, the relu function is used as the activation function.

The second segment includes a convolution layer, an activation function layer, a maximum pooling layer, and a dropout layer. The purpose of the pooling layer is to reduce the input feature map and simplify the network computing complexity; at the same time, feature compression is performed to highlight the main features. We set up the pooling layer by calling MaxPooling2D() function. This function uses the maximum pooling method. This method selects the maximum value of the coverage area as the main feature of the area to form a new reduced feature map. The Dropout layer will consciously reduce the model parameters randomly, making the model simple and reducing overfitting. The input size of the second convolution layer is 64*64*32, the output size is 62*62*32, its output is connected to the second activation function layer, and the output size of the second activation function layer is 62*62*32. Connected to the first largest pooling layer, after 2*2 pooling, the output size is 31*31*32. The maximum pooling layer is followed by the first dropout layer. The purpose of the dropout layer is to reduce overfitting. The set discard ratio is 0.25, and the output size of the first dropout layer is 31*31*32.

The third segment is identical to the first segment and is also composed of a convolutional layer and an activation function layer. The input size and output size of the segment are both 31*31*64. The fourth paragraph is similar to the structure of the second paragraph, except that a flatten layer and a dense layer are added behind the dropout layer. The output size after the fourth activation function layer of the network is 29*29*64, and the output size after the second pooling layer and the second dropout layer is 14*14*64. The function of the flatten layer is to compress the multidimensional data into one. Dimensional data, convenient for the deck layer processing, and finally into the softmax classifier for classification, the output is 22 categories.

IV. Result Analysis

A. Model training and results

The self-built data set has 2200 face images, of which 1540 are used as training sets and 660 are used as verification sets, and the Adam optimizer is used. After 15 rounds of training, batch_size=35 was selected, and each iteration had 44 iterations. The final training error was 0.0532, the training accuracy was 0.9827, the verification error was 0.0215, and the verification accuracy was 0.9785. At this point, the training of the model is completed, and the test is performed on the test set. There are 880 pictures in the test set, and finally the accuracy rate obtained on the test set is 97.63%.

V. Experimental Results

The software environment of this system is a windows system, the programming language used is Python, and Keras is used as a framework. The camera uses a logitech HD1080p camera, which is mounted above the service robot. The service robot performs face recognition during the movement, and the service robot can make subsequent reactions based on the recognition results. Ten people from the self-built database were selected for effect detection. The face recognition effect in a stationary state is shown in Figure 2. At the same time, a face recognition test was performed during the movement of the robot. The recognition effect is shown in Figure 3. The recognition rate is 90% on average when the light environment is good. From the actual results, it can be concluded that the system has a certain anti-interference ability. The system can basically achieve practical results.

Fig. 2. Face recognition of one person

Fig. 3. Face recognition during movement

VI. Conclusion

The face recognition system completed four steps of face collection and preprocessing, model establishment, model training, and real-time recognition. Based on the self-built data set, a convolutional neural network was built and trained. After continuously adjusting parameters, the accuracy rate on the test set was 97.63%. The actual test was performed during the movement of the robot, and the average recognition rate was 90% under good light conditions. The advantages of the face recognition system are that the model is simple, and the memory capacity is small. It is also suitable for practical

application scenarios where the robot moves forward. It can also achieve better recognition results for face recognition within a certain angle. However, the robustness of light needs to be strengthened, and it should be improved in subsequent work.

REFERENCES

[1] Hao N, Liao H, Qiu Y, et al. Face Super-resolution Reconstruction and Recognition Using Non-local Similarity Dictionary Learning Based Algorithm[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(2):213-224.

[2] Jiang J. Face Super-Resolution via Multilayer Locality-Constrained Iterative Neighbor Embedding and Intermediate Dictionary Learning[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2014, 23(10):4220-31.

[3] Aouada D, Al Ismaeil K, Idris K K, et al. Surface UP-SR for an improved face recognition using low resolution depth cameras[C]// IEEE International Conference on Advanced Video & Signal Based Surveillance. 2014.

[4] Zhou K, Zheng L. Multi-pose Face Recognition Based on Improved ORB Feature[J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(2):287-295.

[5] Huang C, Liang Y, Ding X, et al. Generalized joint kernel regression and adaptive dictionary learning for single-image super-resolution[J]. Signal Processing, 2014, 103(C):142-154.

[6] Lee J, Seo Y H. An Efficient Head Pose Determination and its Application to Face Recognition Using Multi-pose Face DB and SVM[C]// Ninth International Conference on Broadband & Wireless Computing. 2015.

[7] Wijaya I G P S, Uchimura K, Koutaki G. Multi-Pose Face Recognition Using Hybrid Face Features Descriptor[J]. 2017.

[8] Yang J. Multi-pose Face Recognition Algorithm Based on Sparse Representation[C]// International Conference on Intelligent Transportation. 2016.

[9] Lee J, Seo Y H. An Efficient Head Pose Determination and its Application to Face Recognition Using Multi-pose Face DB and SVM[C]// Ninth International Conference on Broadband & Wireless Computing. 2015.

[10] Zhang D, Zhang X, Liu H, et al. Image synthesis for sparse representation based multi-pose face recognition[C]// Advanced Research & Technology in Industry Applications. 2014.

[11] Chen H Y, Huang C L, Fu C M. Hybrid-boost learning for multi-pose face detection and facial expression recognition ☆ [J]. Pattern Recognition, 2008, 41(3):1173-1185.

[12] Chen H Y, Huang C L, Fu C M. Hybrid-Boost Learning for Multi-Pose Face Detection and Facial Expression Recognition[C]// IEEE International Conference on Multimedia & Expo. 2007.

[13] Goswami G, Ratha N, Agarwal A, et al. Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks[J]. 2018.

[14] Reale C, Nasrabadi N M, Chellappa R. An analysis of the robustness of deep face recognition networks to noisy training labels[C]// Signal & Information Processing. 2017.

[15] Mohammadi A, Bhattacharjee S, Marcel S. Deeply vulnerable: a study of the robustness of face recognition to presentation attacks[J]. Iet Biometrics, 2018, 7(1):15-26.