Identity Recognition Based on Face Image

Chen Qin\textsuperscript{1,2}, Xiaoobo Lu\textsuperscript{1,2,*}, Pengguo Zhang\textsuperscript{3}, Huibin Xie\textsuperscript{3} and Wenbin Zeng\textsuperscript{3}

\textsuperscript{1} School of Automation, Southeast University, Nanjing 210096, China
\textsuperscript{2} Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China
\textsuperscript{3} Zhejiang Uniview Technologies Co., Ltd, Zhejiang, 310018, China

Email: xblu2013@126.com
*Corresponding Author

Abstract. This article is aimed at designing a human identity recognition algorithm based on face image, which will be used in indoor environment. As the working environment is set as indoor environment, the camera will not be affected much by illumination variation. The key is how to detect human face and handle the variation of facial pose. This article divides the whole recognition process into 4 parts: image pre-processing, face detection, face alignment, feature extraction and comparison. Face detection and feature extraction are the core functions and both realized by deep learning. The process of the whole algorithm can be described as following: images after pre-processing are fed to face detection network to get the locations of face and face landmarks. Then face alignment will be conducted. Finally, deep features of face will be extracted and compared. The unique features of this algorithm are its good performance of handling the variation of facial pose and its clear framework which allows the whole method can be easily adjusted and upgraded.

1. Introduction
As a large number of cameras have been set in public areas and imaged-related researches have got huge progress in recent years, face recognition devices are used in more and more places. Most of the devices which are used for recognizing identity need testers to adjust facial pose to meet recognition requirements, so this kind of devices is not practical on some occasions. This article is aimed at designing a face recognition algorithm which will be used mainly in indoor environment and does not need testers adjust facial pose. The algorithm needs to detect face location automatically and be robust when it handles kinds of facial poses and extract features to recognize identity finally. This algorithm will not interfere testers, so it can be applied to more occasions. For example, it can be used in stations or public transport to help recognize wanted men.

The core techniques of this article are face detection and face recognition. At present there has been many research achievements in the two areas. For face detection, there are Joint Cascade [1], STN [2], Faceboxes [3], SSH [4] and so on. For face recognition, there are Joint Bayesian [5], DeepID3 [6], FaceNet [7] and so on. Most frontier methods of face detection and recognition are based on deep learning, and the excellent performance of deep learning has been approved widely. This article follows the traditional face recognition framework: face detection, face alignment, feature extraction and comparison, and applies different kinds of networks to design a practical face recognition algorithm, supported by some image processing method.
2. Approach
The algorithm can be divided into 4 parts: image pre-processing, face and landmarks detection, face alignment and feature extraction and comparison. Face detection and feature extraction are both based on deep learning. This article mainly researches on how to apply related algorithms to practice.

Pre-processed images are fed to face detection module. After face and landmarks are detected, face alignment will be conducted. Then face features will be extracted and compared with pre-stored features. If the features are similar enough, the identity of current tester will be given. The pre-stored face features also need to be processed by the same four parts before and conserved in the database.

2.1. Image Pre-processing
This part includes several operations: The image size will be limited according to practical requirements to ensure the process time will not be too long. The storage format of images will be changed. Normally the depth of per pixel of one image is 256. The storage format of per pixel can be converted to floating number, because this format suits networks better. Besides, more operation such as filtering and illumination pre-processing can be added according to the working environment and devices.

2.2. Face and Landmarks Detection
This part mainly applied the MTCNN [8] algorithm proposed by Yu Qiao in 2016. This algorithm has showed its excellent performance on face and landmarks detection by cascading 3 convolutional neural networks(CNNs).

This detection algorithm can be briefly summarized in the following outline: Initial images will be scaled to different sizes to get image pyramids; The first CNN P-Net will be used to get many possible candidate windows of faces; Then the second CNN R-Net will refuse most wrong candidate windows; The third CNN O-Net will give the final face windows and facial landmarks position. The face windows produced by 3 CNNs all need to be adjusted by bounding box regression. And non-maximum suppression(NMS) will be employed to merge highly overlapped candidates.

![Figure 1. Structures of 3 face detection networks.](image-url)
input layer; ‘conv’ represents convolution layer; ‘PReLU’ represents Parametric Rectified Linear Unit(PReLU) activation function; ‘pool’ represents pooling layer; ‘InnerProduct’ represents fully connected layer; ‘dropout’ represents Dropout layer; ‘prob’ represents softmax layer; Some number after some layer is the output number of this layer.

The first stage of cascaded network is P-Net. P-Net is a fully convolutional network. Multi-scale images are fed to P-Net and the output of P-Net is divided into 2 branches. One branch produces 2 outputs which can be considered as the probability that current window is a face window. The other branch produces 4 outputs which are the regression parameters of bounding box regression. Candidate windows will be adjusted by these parameters and highly overlapped candidates will be merged by NMS. P-Net will produce a large number of candidates And because P-Net is a fully convolutional network, its result is not convincing enough. The candidates need to be checked by networks which are more powerful.

R-Net and O-Net are similar with P-Net in structure. But they apply fully connected layers to make their outputs more accurate and convincing. Their outputs are also similar with P-Net including follow-up operation like bounding box regression and NMS. O-Net is the final stage of the cascaded network and has an extra output branch, which is the position of facial landmarks. The results produced by O-Net are also the final results of the whole cascaded network. Beside the face window position, facial landmarks position will be used for face alignment.

MTCNN has got excellent performance on frontal face detection in experiments. And it also has got high rate of accuracy in detecting slanted frontal face, side face and other faces which are more difficult be detected. Some detection results are shown in figure 2.

![Figure 2. Some detection results from MTCNN.](a) (b) (c)

To prove the advantage of MTCNN, this article conducted an experiment to compare MTCNN with the previous CNNs [9]. They are tested on the same data set which includes frontal faces and side faces. The result is shown in table 1.

| Face Pose | Algorithm | Accuracy |
|-----------|-----------|----------|
| Frontal Face | [9] | 93.33% |
| Side Face | MTCNN [9] | 92% |

The data in table 1 has shown that MTCNN has better performance on the accuracy of detecting faces. The difference is more obvious on the side face data set. And it can prove that MTCNN is robust in detecting faces with multiple poses.

2.3. Face Alignment

As the position of face window and facial landmarks has been got, face alignment can be conducted based on these information. The purpose of face alignment is to adjust all face poses to one standardized pose. Alignment can make the algorithm more robust in recognizing faces and reduce the influence from face poses as much as possible.
The concrete operation is to rotate the image. The center of rotation is the midpoint of the line segment between two eyes. The rotation angle is the angle between the line segment between two eyes and the horizontal line of the image. Briefly speaking, the purpose of alignment is to make two eyes are on the same height. Affine transformation can be applied to complete the operation. Because the transformation does not include scaling or other operations, the transformation can be written as the following:

\[
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & -\sin \theta & c1 - c1 \ast \cos \theta + c2 \ast \sin \theta \\
\sin \theta & \cos \theta & c2 - c1 \ast \sin \theta - c2 \ast \cos \theta \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]  

In equation (1), \( \theta \) is the rotation angle. \((c1, c2)\) is the coordinate of the midpoint. \((u,v)\) is the coordinate of the corresponding point in the destination image, as \((x,y)\) is the coordinate of the corresponding point in the source image.

After rotation, only the face part of the image will be conserved. The concrete operation is based on human face structure.

Some face alignment results are shown in figure 3.

![Figure 3. Some face alignment results.](a) (b) (c) (d)

2.4. Deep Features Extraction and Comparison

For getting the deep features which can really describe face, one deep network which is capable enough of distinguishing different faces is necessary. So this article has chosen VGG16 [10] as the deep network to extract face features and adjusted the net structure to get more detailed deep feature.

The net structure used in this article is shown in table 2:

| Parts of the Net                  | Components                          |
|-----------------------------------|-------------------------------------|
| Input Layer                       | Input(224*224 RGB)                  |
| Convolutional Layers_1            | conv3-64                            |
| Convolutional Layers_2            | conv3-128                           |
| Convolutional Layers_3            | conv3-256                           |
| Convolutional Layers_4            | conv3-512                           |
| Convolutional Layers_5            | conv3-512                           |
| Fully Connected Layer_1           | fully connected-4096                |
| Fully Connected Layer_2           | fully connected-4096                |

In table 2, conv3-64 indicates that this convolution layer’s kernel size is 3*3 and output number is 64. The net has 5 convolution parts. Every part’s final layer is a max pooling layer. The last two layers are fully connected layers and their output numbers are both 4096. And the output vector from the last layer is the deep feature vector of face.

VGG net is improved from AlexNet [11], it is deeper than AlexNet and uses smaller convolution kernel. It is widely approved that deeper net has better description ability. VGG is trained partially because of its structure and it makes VGG has good performance on extracting features in many applications.
As the features have been extracted, the next step is to compare features with the pre-stored features. This article apply cosine similarity to describe the similarity between two feature vectors. The closer to 1 the cosine similarity is, the more similar two vectors are. If two feature vectors are similar enough, the most similar one will be given as the recognizing result. It needs to be noticed that the pre-stored features must be achieved from the same operations.

3. Experiments

To test the algorithm proposed in this article, 30 testers took part in the experiment. For every tester, 5 reference face images of different visual angle are conserved. These pictures include 5 visual angles: front, left side, right side, look up, overlook. These images have been processed by the operations mentioned before and their features are conserved in the database for comparison.

This experiment needs to take several videos of testers with a HD camera and videos are required to take multiple visual angles. Then some frames from the videos are intercepted as test samples. Here is the description of test samples: The number of samples is 600. These samples can be divided into 4 groups by visual angle: front: 150; side: 150(including left and right); overlook: 150; look up: 150.

This experiment not only wants to test the algorithm’s performance on faces with different visual angles, but also wants to observe the influence of face alignment to the accuracy. Besides, this experiment also researched on how the similarity threshold affects the mis rate and mistake rate.

When similarity-threshold equals 0.6(That is to say, if the similarity is larger than 0.6, the current face may belong to the reference tester.), the test result is shown in table 3.

|                  | With alignment | Without alignment |
|------------------|----------------|------------------|
| Front            | 94.67%         | 86.67%           |
| Side             | 89.33%         | 86.67%           |
| Overlook         | 84%            | 81.33%           |
| Look up          | 84.67%         | 80.67%           |
| All samples      | 88.17%         | 83.83%           |

It can be observed from table 3 that the accuracy of frontal samples is highest probably because frontal samples can carry more information. And face alignment is significant for improving accuracy especially for the frontal samples. Face alignment actually makes the algorithm more robust in dealing with different visual angles.

The test before is under the situation that similarity-threshold equals 0.6. Another test was conducted when similarity-threshold was set 0.8 for comparison. And the test result is shown in table 4.

|                  | Miss Rate | Mistake Rate | Total   |
|------------------|-----------|--------------|---------|
| Similarity-threshold=0.6 | 4.5%      | 7.33%        | 11.83%  |
| Similarity-threshold=0.8  | 8.33%     | 3.83%        | 12.17%  |

Table 4 shows that when similarity-threshold is increased, miss rate increases and mistake rate decreases. This result is within expectation. Although the increase of similarity-threshold causes the total error rate to increase, it increases the accuracy of the recognition of frontal examples from 94.67% to 96.67%. Therefore, similarity-threshold can be set depending on the distribution of samples. For example, if the frontal samples take the majority in practical application, the increase of similarity-threshold will benefit the accuracy.

The runtime of the whole process increases linearly with the number of faces detected in the image. All the images are limited to be smaller than 1000*1000 and the program is run on GTX1060. The
average runtime for each face is 80ms. Some recognition results are shown in figure 4. On the left-top of the face window is the name of the tester and the similarity with pre-stored faces.

![Recognition Results](image)

Figure 4. Some recognition results.

4. Conclusion
This article proposes an identity recognition algorithm which contains face detection, face alignment and feature extraction. Its core functions are realized by deep networks. The experiments have indicated that this algorithm performs well on recognizing faces with various poses in the indoor environment. The framework is simple and functions can be divided clearly. If one of the 4 parts has got better method, it is easy to replace the old one and improve the performance. For example, the current operations in alignment part are mainly applied to handle the skew of frontal face, but these operations can’t improve the performance in other situations as much as frontal faces. Therefore, Other alignment operations can be added if it is necessary.

5. Acknowledgments
This work was supported by the National Natural Science Foundation of China (No.61871123), Key Research and Development Program in Jiangsu Province (No.BE2016739) and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

6. References
[1] Chen, D., Ren, S., Wei, Y., Cao, X., & Sun, J. (2014, September). Joint cascade face detection and alignment. In European Conference on Computer Vision (pp. 109-122). Springer, Cham.
[2] Chen, D., Hua, G., Wen, F., & Sun, J. (2016, October). Supervised transformer network for efficient face detection. In European Conference on Computer Vision (pp. 122-138). Springer, Cham.
[3] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017, October). Faceboxes: A CPU real-time face detector with high accuracy. In 2017 IEEE International Joint Conference on Biometrics (IJCB) (pp. 1-9). IEEE.
[4] Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). Ssh: Single stage headless face detector. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4875-4884).
[5] Chen, D., Cao, X., Wang, L., Wen, F., & Sun, J. (2012, October). Bayesian face revisited: A joint formulation. In European Conference on Computer Vision (pp. 566-579). Springer, Berlin, Heidelberg.
[6] Sun, Y., Liang, D., Wang, X., & Tang, X. (2015). Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873.
[7] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).
[8] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.
[9] Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5325-5334).
[10] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).