A Real-time Face Tracking and Recognition System Based on SeetaFace

Dilizhati Yilihamu, Palidan Tuexun, Abdusalam Dawut and Askar Hamdulla*
School of Software, Xinjiang University, Urumqi, 830046, China
*Corresponding Author Email: askar@xju.edu.cn

Abstract. A real-time face tracking and recognition system is constructed based on Seetaface, and it is proposed to use Gamma correction to reduce the impact of illumination changes on the detection and recognition results. First, the system has face detection function, then by using OpenCV to get the image in front of the camera, the system completes the function of real-time face tracking, users can also upload local images to make face comparison. If a stranger's face is detected, the system can collect information and train it. The system achieves multiple face tracking as well as recognition. Through the analysis and evaluation on public datasets, the improved face tracking and recognition system has a satisfactory accuracy and performance.

1. Introduction
In recent years, the field of artificial intelligence has developed rapidly. It is urgently needed to be applied in smart home, security system, industrial automation, medicine, finance and other fields. The user groups range from individuals to enterprises to industries. Face recognition technology plays an important role.

This paper is built based on the SeetaFace engine, which is a face recognition algorithm designed and developed using the C++ code. The advantage of this model is that it reduces the dependence on third-party library functions. It was conceived and developed by the R&D group of Mr. Shiguang Shan, the Chinese Academy of Sciences, and is now open-source, so that we can use the SeetaFace face recognition model for academic research[1]. In order to improve the accuracy of face recognition and tracking in multiple environments, the system has been improved by proposing to use a gamma correction algorithm to preprocess face images, which is a significant improvement compared to the original system.

2. System Architecture
The system is composed of three main parts. The first part is to carry out dynamic real-time recognition. Turn on the computer's camera for tracking and recognition. The second part is that the user can read the local picture from the computer, which can realize the function of face comparison. If the face to be recognized is a stranger's face, then the third part of the system will be used for training, intercepting the stranger's face picture, and then let the system complete the training automatically. As shown in the figure below.
2.1. Preprocessing Based on Gamma Correction

The system first improves the overall brightness of the image by means of an adaptive Gamma correction, which is a non-linear operation on the input image grayscale values so that the output image grayscale values are exponentially related to the input image grayscale values [2]. We use the Gamma correction algorithm in order to cope with the illumination imbalance in digital images.

\[
O(i, j) = 255 \left\lfloor \frac{I(i, j)}{255} \right\rfloor \gamma
\]

In this formula, \(I(i,j)\) is the input image, the output image is \(O(i,j)\), and \(\gamma\) is the control parameter. When the control parameter \(\gamma\) is greater than 1, our image will become brighter. When the control parameter \(\gamma\) is less than 1 and greater than 0, our image will become darker. The algorithm for \(\gamma\) less than 0 is meaningless.

This algorithm can obtain satisfactory results by adjusting the parameter \(\gamma\) when the overall image is darker or overall brighter. However, if there are underexposed or overexposed areas in the image at the same time, the same parameter cannot obtain satisfactory results at the same time. Therefore, an algorithm that changes \(\gamma\) with the local area information of the image can be introduced to obtain a more satisfactory effect. A commonly used form of this formula is as follows:

\[
O(i, j) = 255 \left\lfloor \frac{I(i, j) I(i, j, N(i, j))}{255} \right\rfloor \gamma
\]

However, we still have the problem of over-correction when using this formula. Some color pictures may be lost. Therefore, the author has further corrected this formula in the literature [3], and adjusted it mainly in two aspects, and the formula is shown below.

\[
\gamma[i, j, N(i, j)] = \alpha \frac{128 - B_{mask}(i, j)}{128}
\]

When the average value is greater than 128, it means that the point is a dark pixel and the surrounding area is also a dark pixel, so \(\gamma\) needs to be less than 0 in order to brighten it up, using the following formula.

\[
\alpha = \frac{\ln(0.5)}{\ln(128/5)}
\]

When the overall average value of the image is less than 128, the corresponding description is that the current point is a brighter pixel and the surrounding pixels are also brighter. If the mask value is
128, there will be no change. At the same time, the farther the mask value is from 128, the correction amount The bigger it is, and another feature is that there will be no change between pure white and pure black. Use the following formula to calculate.

\[ \alpha \approx \frac{\ln(I/255)}{\ln(0.5)} \]  

(5)

The degree of correction is reflected by \( \gamma \). The figure below shows the result we got by performing Gamma correction.

![Gamma correction effects](image)

**Figure 2.** Gamma correction effects

### 2.2. Face Detection Module

In this paper, the face recognition system based on deep learning also uses a cascade structure in the face detection stage, which is designed for faces with different poses. The funnel-shaped cascade structure FuSt (the cascade of the funnel structure) is designed from top to bottom, from coarse to fine [4]. FuSt (Funnel-Structured Cascade) is a funnel-shaped cascade structure. The cascade is divided into three levels in total, and for a face image, it gradually distinguishes between human and non-human face regions from top to bottom. The following figure shows the implementation flow of the funnel cascade structure based on FuSt.

![FuSt Cascade Structure](image)

**Figure 3.** FuSt Cascade Structure

### 2.3. Facial Feature Location Module

The SeetaFace Alignment module uses a coarse-to-fine auto-encoder network (Coarse-to-Fine Auto-encoder Networks, CFAN) when locating facial feature points [6]. The CFAN method uses the location of 5 key facial feature points in the facial feature point positioning module. These five key face points are also the most prominent features of the face. The pictures are labeled with five key feature points of the face. In order to improve the speed and efficiency of positioning, the SeetaFace Alignment module reduces the number of CFAN cascades to two levels. Without face detection, the processing time for each face reached 5ms through such optimization. The following figure shows the main process of locating the five key feature points in the face image by using SeetaFace Alignment.
2.4. Face Recognition Module

The feature extraction of this module is based on the deep network model VIPLFaceNet, which has a deep network structure with 7 convolutional layers and 3 fully connected layers [8]. VIPLFaceNet replaces 11*11 size convolution with 9*9 size convolution in the first layer compared to AlexNet, and removes the LRN layer from the network because of this localized The response normalization layer is not very helpful for us to initialize the parameters. In this model not only the number of feature maps in the convolutional layer is reduced, but also the number of nodes in the FC2 layer. To speed up the convergence of this model, a Fast Normalization Layer (FNL) is also used to improve the generalization capability of VIPLFaceNet [9]. The following figure is the process of real-time face recognition system.

3. System Implementation and Analysis

Our face tracking and recognition system has a total of three modules. First, we can turn on our computer’s camera for real-time tracking and recognition. Secondly, we can turn on the photos on our computer for face detection and recognition. Finally, we can train stranger face on the system. The following figure shows the main interface of the system.
For faces of strangers, we need to collect face information, which can be successfully recognized through training. We first click the "Collect" button to enter the collection information page, each tester takes 7 to 8 screenshots, and then click the training button, and the system will start training facial images. The following figure shows the facial image after training through the system.

![System interface for face recognition](image)

**Figure 7.** System interface for face recognition

The accuracy of facial feature positioning directly affects the subsequent recognition process. We conducted experiments using different poses of the face in the dataset. The result is shown in the figure below.

![Gamma Facial feature location](image)

**Figure 8.** Gamma Facial feature location

We quantitatively evaluated our system on the AFLW dataset with six metrics, LE: left eye, RE: right eye, N: nasal tip, LM: left corner of the mouth, RM: right corner of the mouth, and Average: normalized according to the distance between the centers of the two eyes. The comparison is shown in the figure below.

![Localization errors and comparisons on the AFLW dataset](image)

**Figure 9.** Localization errors and comparisons on the AFLW dataset
In order to see more clearly and concretely the practical application effect of this system, we compare the accuracy of its face recognition with the traditional face recognition system. In order to ensure the accuracy of the experiment, we conducted the tests on LFW dataset and CASIA-FaceV5 dataset respectively, and the comparison is shown in the following table.

| Dataset       | FaceNet | SeetaFace | DeepID1 | Our System |
|---------------|---------|-----------|---------|------------|
| LFW           | 93.4    | 97.1      | 97.45   | 97.8       |
| CASIA-FaceV5  | 93.1    | 94.2      | 94.4    | 95.1       |

4. Conclusions
In this paper, a multiple face real-time tracking and recognition system with automatic detection, tracking and recognition functions is designed on the development platform of Visual Studio 2013 + OpenCV, and on this basis, the system functions such as face acquisition, face training and debugging are realized. In this paper, we designed and developed a real-time face tracking and recognition system, and after analysis, I found that the system still has some shortcomings. The difficulty of performing facial feature point localization. In my future research and study, I will further explore the above problems and optimize the response speed of the whole system in an effort to find an appropriate solution.

5. Acknowledgments
This work was supported by Natural Science Foundation of China (61662076).

6. References
[1] Hu G , Xiao Y , Cao Z , et al. Towards Real-time Eyeblink Detection in The Wild: Dataset, Theory and Practices[J]. 2019.
[2] L. M. I. Leo Joseph, S. Rajarajan. Reconfigurable hybrid vision enhancement system using tone mapping and adaptive gamma correction algorithm for night surveillance robot[J]. Multimedia tools and applications, 2019.
[3] Singh H , Kumar A , Balyan L K , et al. Fractional-Order Integration based Fusion Model for Piecewise Gamma Correction along with Textural Improvement for Satellite Images[J]. IEEE Access, 2019:37192-37210.
[4] Shuzhe Wu, Meina Kan, Zhenliang He, Shiguang Shan, and Xilin Chen. Funnel-Structured Cascade for Multi-View Face Detection with Alignment-Awareness. Neurocomputing (under review), 2016.
[5] Yan S,Shan S,Chen X,et al.Locally Assembled Binary (LAB) feature with feature-centric cascade for fast and accurate face detection[C]//IEEE Conference on Computer Vision & Pattern Recognition.IEEE,2008.
[6] Jie Zhang, Shiguang Shan, Meina Kan, Xilin Chen. Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment. ECCV 2014.
[7] Zhanpeng Zhang, Ping Luo, Chen Change Loy, Xiaou Tang. Facial Landmark Detection by Deep Multi-task Learning. ECCV 2014.
[8] Liu X,Kan MWanglong W U,et al.VIPLFaceNet: an open source deep face recognition SDK[J].Frontiers of Computer Science,2017,11(2).
[9] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. CVPR 2015.
[10] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaou Tang. From Facial Parts Responses to Face Detection: A Deep Learning Approach. ICCV 2015.