Face Recognition System for Complex Surveillance Scenarios

Xu Guo, Jisheng Nie

Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing, China

guoxu9266@163.com

Abstract. In recent years, with the continuous development of the Internet and artificial intelligence, face recognition technology has also been widely used in many application scenarios. Facing complex surveillance scenarios, face recognition technology still faces great challenges. This paper focuses on implementing real-time and efficient face recognition systems in complex surveillance scenarios, such as insufficient lighting, small faces, dense crowds, and sides at 45 ° environment. The system is mainly based on RetinaFace for face detection and face alignment, and uses lightweight mobilenet (0.25) as the backbone network of RetinaFace. Facial feature extraction is based on deep residual neural network combined with ArcFace loss, and feature matching is performed by Euclidean distance. The experimental results show that the face recognition system has good real-time performance, accuracy and robustness.

1. Introduction

Nowadays, surveillance equipment is widely used, and systems based on face recognition technology are also widely used and applied. Face recognition technology has been applied to many scenarios. With the continuous popularization of monitoring equipment, face recognition technology still faces great challenges in the face of complex monitoring scenarios. How to efficiently process surveillance video to extract useful information has become an important research direction today. By applying face recognition technology to video surveillance to improve the efficient processing of video, security work and human life will be greatly improved. Facing complex surveillance scenarios, face recognition systems need to detect and recognize faces in video surveillance in real time efficiently. This paper studies the related algorithms of face recognition and implements face recognition in complex surveillance scenarios. The system is mainly composed of cameras, network equipment, back-end server, face recognition platform software and other equipment. Obtain video images through surveillance cameras, upload videos to the server, read video frames from the video stream through OpenCV, and faces are detected by RetinaFace. Faces are captured and saved according to the detected face frame coordinates, and then deep residual neural networks are combined with ArcFace loss to extract features from the faces, and then retrieve whether the faces are in the feature database. The system has good real-time performance, high accuracy and robustness, and can be applied in many application scenarios.

2. RetinaFace

RetinaFace[1] is a robust single-level face detector, which performs pixel-wise face localisation on various scales of faces by taking advantages of joint extra-supervised and self-supervised multi-task learning. RetinaFace is a face detection algorithm that emerged in May 2019 and achieved state-of-
the-art at that time. At present, RetinaFace ranks second only with extremely narrow accuracy difference. RetinaFace improves the single-stage face detection framework and propose a state-of-the-art dense face localisation method by exploiting multi-task losses coming from strongly supervised and self-supervised signals. By manually annotating five face landmarks on a WIDER FACE dataset, significant improvements in hard face detection were observed with the assistance of this extra supervision signal. On the WIDER FACE hard test set, RetinaFace outperforms the state of the art average precision (AP) by 1.1% (achieving AP equal to 91.4%).

RetinaFace network architecture as shown in figure 1.

Figure 1. RetinaFace network architecture.

RetinaFace loss function diagram as shown in figure 2.

Figure 2. RetinaFace loss function diagram.

Multi-task Loss as follows:

\[ L1 = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel}. \]  

\[ L_{cls}(p_i, p_i^*) \] represents face classification loss, \( p_i \) is the predicted probability of anchor \( i \) being a face and \( p_i^* \) is 1 for the position anchor and 0 for the negative anchor. \( L_{box}(t_i, t_i^*) \) represents face box regression loss, \( t_i = \{t_x, t_y, t_w, t_h\}_i \) and \( t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\}_i \) represent the coordinates of the predicted box and ground-truth box associated with the position anchor. \( L_{pts}(l_i, l_i^*) \) represents facial landmark regression loss. \( l_i = \{l_{x1}, l_{y1}, ..., l_{x5}, l_{y5}\}_i \) and \( l_i^* = \{l_{x1}^*, l_{y1}^*, ..., l_{x5}^*, l_{y5}^*\}_i \) represent the predicted five facial landmarks and groundtruth associated with the position anchor. Similar to the box centre regression, the five facial landmark regression also employs the target normalization based on the anchor centre. \( L_{pixel} \) represents dense regression loss.

3. ArcFace

ArcFace[2] is easy and efficient. ArcFace achieves state-of-the-art performance on ten face recognition benchmarks including large-scale image and video datasets.

The formula for Softmax Loss[3] is as follows:
\[ L_2 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{W^T_{y_i}x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W^T_{j}x_i + b_j}} \]  

Softmax Loss does not explicitly optimize features to positive samples with higher similarity, negative samples can have lower similarity[4]. That is, does not expand decision boundaries, so the offset \( b_j \) is then set to 0. \( W^T_{j}x_i \) transform into \( ||W_j|| \cdot ||x_i|| \cos \theta_j \). \( \theta_j \) is the angle between the weight \( W_j \) and the feature \( x_i \). By normalization, the modulus of \( W_j \) and \( x_i \) is fixed to 1, and re-scale it to s, then the above loss can become:

\[ L_3 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s \cos(\theta_{y_i})}}{e^{s \cos(\theta_{y_i})} + \sum_{j=1,j \neq y_i}^{n} e^{s \cos \theta_j}} \]  

In order to enhance the intra-class compactness and inter-class discrepancy, add an additive angular margin penalty \( m \) between \( x_i \) and \( W_{y_i} \). Additive Angular Margin Loss(ArcFace Loss) as follows:

\[ L_4 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1,j \neq y_i}^{n} e^{s \cos \theta_j}} \]  

4. Design and Implementation of Face Recognition System

4.1. Face recognition system design.

Face recognition is divided into four processes: face detection, face alignment, feature extraction and feature matching. This paper studies the related algorithms of face recognition, and achieves face recognition in complex surveillance scenarios. The system is mainly composed of camera, network equipment, back-end server, face recognition platform software and other equipment. The B / S mode is mainly used. Users can view the surveillance video information of the surveillance camera in real time on the web page. The front-end part uses html + js + css to implement business logic and pages. The back-end server mainly includes video storage, image storage, face recognition, etc. Real-time capture of face pictures on the videos obtained by surveillance cameras. The captured picture can be compared with the list library in real time. When the similarity reaches the threshold, the person can be determined. Enter a face picture, extract the facial feature vector, and search according to the feature comparison algorithm and the snapshot library to help users find relevant personnel. Video images are obtained by monitoring cameras, the video is uploaded to the server, video frames are read from the video stream through OpenCV, and faces are detected by RetinaFace. The face is intercepted and saved according to the detected face frame coordinates, and ArcFace is used to extract feature from the face picture, and then retrieve whether the face is in the feature database. The process of image face recognition system is shown in Figure 3.

Figure 3. System processing flow chart.
4.2. Experimental process

4.2.1. Read video stream data. The read video comes from the uploaded video file or the real-time video stream collected by the camera. The method of obtaining video can be changed by changing the function parameters. Read the video frames in the video stream by calling OpenCV.

4.2.2. Face detection and face alignment. Face detection is mainly to capture the presence of faces in the video stream in a dynamic and complex environment, determine the specific location, separate the face from it, and mark the area of the face. Face detection and face alignment based on RetinaFace, using mobilenet (0.25) as the backbone network of RetinaFace, the detection speed is fast, while detecting every other frame. Face detection also includes face detection on the input image to find all the faces in the image. Then use OpenCV to intercept the detected face to obtain face images. Face detection results is shown in Figure 4.

![Figure 4. Face detection results.](image)

4.2.3. Face feature extraction. Feature extraction, as the most critical step of face recognition, is more likely to extract feature unique to the face, which plays an important role in feature matching. Our network and model bear the heavy task of feature extraction. Excellent network training strategies make the model more robust. However, the performance of the Resnet network is good, in order to improve the performance model of face recognition, in addition to optimizing the network structure, it is another option to modify the loss function, which can make the model learn more valuable information from the existing data.

In previous contact with the classification problem of a large part of using the Softmax loss as the loss of the network layer, experiments show that Softmax loss considering whether the samples can be classified correctly, and in expanding class and narrow the distance between the similar between heterogeneous samples within the class of distance between sample has a lot of optimization on the question of space. An improved neural network architecture that combines deep residual neural networks and ArcFace loss. ArcFace Loss replaces the Loss function Softmax of deep residual neural network. A 512-dimensional feature vector is extracted from the input image through a pre-trained model.

4.2.4. Face feature matching. Compare with the features of the feature library. Set the threshold, calculate the Euclidean distance between the current feature and the feature library feature, compare
with the threshold value, and return the person corresponding to the feature if it is smaller than the threshold.

The system designed in this paper uses the Euclidean distance algorithm, which is simple and efficient, and is used by various image recognition algorithms. Euclidean distance is the easiest to understand intuitively in distance measurement methods. Euclidean distance is the most widely used distance definition in the field of image processing. It is defined as the true distance between 2 points in m-dimensional space, or the natural length of the vector (the distance from the point to the origin). After getting 512-dimensional vectors through deep residual neural network combined with ArcFace loss, you can compare the similarity of the two vectors. Euclidean distance is the easiest to understand intuitively in distance measurement methods, efficient and simple. This paper uses Euclidean distance to calculate and process two 512-dimensional vectors, and then judges whether the face is a person in the database. The two facial feature formulas are shown below:

\[ p_1 = (x_1, x_2, x_3, \ldots, x_n) \]  
\[ p_2 = (y_1, y_2, y_3, \ldots, y_n) \]

The Euclidean distance is calculated with equation (7) as follows:

\[ d_{(x,y)} = \left( \sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2} \]  

The calculated corresponding value is compared with the discrimination threshold, and if it is smaller than the discrimination threshold, it can be determined as the same person; if it is greater than the discrimination threshold, it is not the same person. Face recognition results are shown in Figure 5.

5. Conclusion
Face recognition is affected by factors such as light, angle, and number of faces. In order to better improve the detection effect and real-time performance of the face recognition system, this paper makes in-depth research on face detection and face feature extraction algorithms, and proposes a face recognition system design scheme in complex monitoring scenarios. Face detection and face alignment based on retinaface use lightweight mobilenet (0.25), which has good detection effect on small face and good real-time performance. Deep residual neural network combined with ArcFace loss is used for facial feature extraction. It has good robustness in complex surveillance scenarios. The system can achieve real-time face recognition. This system can be applied to railway transportation,
community security and attendance, etc. At the same time, the system also has the meaning of continuous improvement, and there is a lot of room for improvement.

References
[1] Deng J.K, Guo J, Zhou Y.X, Yu J.K, Zafeiriou S. (2019) RetinaFace: Single-stage Dense Face Localisation in the Wild. In: IEEE Computer Vision and Pattern Recognition. Long Beach, CA.
[2] Deng J.K, Guo J, Zafeiriou S. (2018) ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In: IEEE Computer Vision and Pattern Recognition. USA.
[3] Wang F, Liu W, Liu H, Cheng J. (2018) Additive margin softmax for face verification. In: IEEE Signal Processing Letters. 2, 3, 7
[4] Chen Z, Ding P. (2019) Face recognition based on improved residual neural network. In: Chinese Control and Decision Conference. Nangchang. 4626-4629.