3D Face Recognition Based on Twin Neural Network Combining Deep Map and Texture

Kangming Xu, Xianmei Wang* and Zhenghua Hu
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
e-mail: xmwang@ustb.edu.cn

Zihao Zhang
Center of AI and Intelligent Operation R&D
China Mobile Research & Institute
Beijing, China
e-mail: zhangzihao613@163.com

***Abstract*—Massive amount of training samples is a challenge for 3D face recognition using deep learning frame. This paper shows a method that uses deep twin neural network for 3D face recognition by blending face 3D depth and 2D texture. First, a depth map is generated. In order to repair holes in the 3D face model with low complexity, we map those 3D hole points into 2D plane, and then reverse them back to 3D space by the least square rule. Second, a convolution kernel model with two layer channels is used to fuse face image and depth image. Finally, after sample pairs are generated, 3D face recognition is performed by convolutional twin neural network. The experimental results on CASIA-3D dataset show that, compared with the classical CNN method, the recognition accuracy of our method increases about 2.85%. And in the case of using small training sets, the recognition rate of our method is about 4% higher.**

Keywords-3D face recognition; convolutional neural network; twin neural network; feature fusion

I. Introduction

Face recognition has been an important research and application focus in the field of computer vision in the last twenty years [1], [2]. However, there are still many challenges for face recognition in unconstrained environment, including complex background, illumination variation, posture changing and occlusion, etc.

Compared with 2D (two-dimensional) face, 3D (three-dimensional) face has extra spatial information. Therefore 3D face detection can gain higher robustness and accuracy, especially in the unconstrained environment. At present, 3D face recognition methods mainly use model matching in 3D face space [3-6].

At present, deep learning has become the main approach for 2D face recognition and achieved excellent effect. However, due to the complexity of the point cloud in 3D face model, it is usually very hard to directly apply deep learning to 3D face recognition. In addition, large-scale acquisition of 3D faces and hole repairing are also very difficult tasks. Therefore, how to obtain a high-quality 3D model and how to train a recognizer in the case of small sample size are both the urgent problems in 3D face recognition research based on deep learning.

Aiming at resolving the issue of 3D face recognition with small-scale samples, this paper proposes a face recognition algorithm using CNN-based (convolutional neural network) twin neural network, shown in Fig. 1.

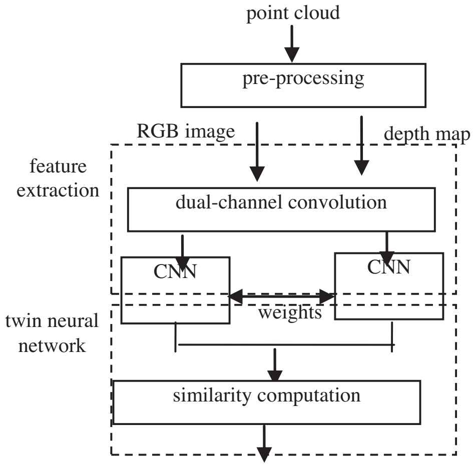


Figure 1. The structure of our method.

II. 3D Facial Model Pre-processing

The purpose of the pre-processing stage is to obtain a grayscale depth map reflecting the facial depth information. This stage is roughly divided into three sub-steps, namely, hole filling, data normalization, and grayscale depth map generation based on cubic interpolation. Before hole filling, reference edge linkage is carried out.

A. Face Hole Repairing Based on 2D Feature Plane

In order to reduce the influence of holes on depth information, this paper proposes a method to reconstruct 3D hole surface based on 2D plane. The basic repairing principle is to first extract the 3D hole edge point, and use the least squares method to fit the three-dimensional surface expression at the hole; then project the hole edge point onto the 2D plane, The hole projection on the 2D plane is uniformly meshed, the coordinates of each grid point are calculated, and the filled 2D grid points are mapped back to the 3D space. After adding the new points into the original 3D point cloud, re-triangulates the triangle and reconstructs a new face model.

978-1-7281-0535-2/19/$31.00 ©2019 IEEE

Assume $p_i(x_i, y_i, z_i)$ is an edge point in a 3D hole, and $p_i'(u_i, v_i)$ is its corresponding points after 2D projection. Then the hole repairing process based on 2D projection plane can be described as several sub-processes.

(1) Hole surface fitting based on 2D points by LSE Rule

Suppose the relationship of z_i to x_i and y_i is a polynomial function shown in (1).

$$z_i(x_i, y_i) = ax_i^2 + bx_iy_i + cy_i^2 \tag{1}$$

The total squared error Q^2 generated by the fitting of all 3D hole edge points is defined as

$$Q^2 = \sum_{i=0}^{n}(ax_i^2 + bx_iy_i + cy_i^2 - z_i)^2 \tag{2}$$

The best parameters a, b and c in (2) could be obtained by minimizing $\boldsymbol{Q^2}$, which means the gradient at minQ^2 is zero. Derive function (2) with respect with a, b and c, the derivation equation set is expressed as (3).

$$\begin{cases} \dfrac{\partial Q^2}{\partial a} = \sum_{i=0}^{n} 2x_i^2(ax_i^2 + bx_iy_i + cy_i^2 - z_i) \\ \dfrac{\partial Q^2}{\partial b} = \sum_{i=0}^{n} 2x_iy_i(ax_i^2 + bx_iy_i + cy_i^2 - z_i) \\ \dfrac{\partial Q^2}{\partial c} = \sum_{i=0}^{n} 2x_iy_i^2(ax_i^2 + bx_iy_i + cy_i^2 - z_i) \end{cases} \tag{3}$$

The optimal parameters a_0, b_0 and c_0 are calculated by setting above derivation equations zero.

(2) Projection from 3D space to 2D plane

Equation (4) shows how to map a 3D point $p_i(x_i, y_i, z_i)$ into 2D plane.

$$\begin{cases} u_i = x_i \\ v_i = y_i \end{cases} \tag{4}$$

(3) Interpolation of grid points in 2D plane

First we divide the 2D hole area into grids, and then calculate the coordinates of each grid intersection point. Let $v_{\max}$ and $v_{\min}$, $u_{\max}$ and $u_{\min}$ respectively represent the maximum and minimum values of the hole edge points after projection, $\overline{d}$ be the mean length of all triangle edges in 3D space. The number of horizontal and vertical lines is calculated by (5).

$$\begin{cases} l_u = (u_{\max} - u_{\min}) / \overline{d} \\ l_v = (v_{\max} - v_{\min}) / \overline{d} \end{cases} \tag{5}$$

If $(i \in [0, l_u], j \in [0, l_v])$ indicates the grid point number along horizon and vertical direction, then the coordinate of a grid point $P'(u_{(i,j)}, v_{(i,j)})$ has following form.

$$\begin{cases} u_{(i,j)} = u_{\min} + \dfrac{u_{\max} - u_{\min}}{l_u} \times i \\ v_{(i,j)} = v_{\min} + \dfrac{v_{\max} - v_{\min}}{l_v} \times j \end{cases} \tag{6}$$

(4) Reverse *mapping and generation of new model*

According to function (6), the grid points in the 2D projection plane are mapped one by one into the 3D space to form a 3D hole surface. Finally, The new 3D hole points together with the ones after edge linkage, are re-triangulated to form a new face model.

$$\begin{cases} x = u \\ y = v \\ z = au^2 + buv + cv^2 \end{cases} \tag{7}$$

III. Face Feature Extraction based on CNN with Dual-channel Convolution Kernel

In our method, CNN architecture adopts LeNet5. To decrease the computation burden, we fuse the deep channel and the texture channel into one channel based on a simple convolution kernel.

A. *Input Fusion byt Convolution Kernel*

In order to simplify the input layer structure of the neural network, a two-channel convolution kernel model (shown in Fig. 2) is constructed to primarily fuse the depth map and gray image together. This structure can synchronously traverse the two-dimensional face texture image and the face depth image to generate a fused convoluted feature map by adding two convolution results together, which will serve as the input of subsequent CNN to generate different deep features. In our method, the size of the two-channel convolution kernel is 5 × 5.

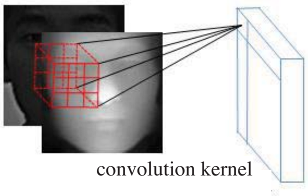


Figure 2. Input fusion by convolution kernel.

B. *Feature Extraction by CNN*

Our CNN-based feature extraction network includes an input layer, a four-layer convolution layer, a four-layer pooling layer and a fully connected layer. The structure is shown in Fig. 3.

We construct different convolution kernels for filtering with 4 convolution layers and 4 down-sampling layers. In turn, the kernel number, size and slide stride for convolutional kernels are set as (64, 128, 256, 512), (1,1,1,2) and (5*5, 3*3 3*3 and 3*3). Moreover, to avoid border effect, all the convolutional layers use border supplement effect. As the pool layers, all use maximum pooling mode, receptive fields of 3x3 and stride of 2.

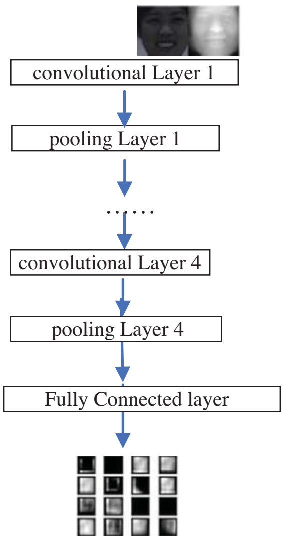


Figure 3. The structure of feature extraction network.

Avoiding few training samples or over-training, the full connection layer applies the Dropout mechanism.

In order to make the parameters in the neural network model move closer to the local optimal solution and avoid the loss of the gradient, the Gaussian random distribution is applied to set the initial parameters, and the loss function employs cross entropy.

IV. Face Recognition Based on Twin Neural Network

Compared with the traditional deep learning architecture, the convolutional neural network owns the advantages of simple structure, strong feature extraction, fast training efficiency and low training cost. However, in the case of small scale of samples, the convolutional neural network will lead to overfit. Therefore, we proposes an improved CNN structure combining the convolutional neural network with the twin neural network.

The structure of the twin neural network based on CNN is shown in Fig. 4. It can be divided into two parts: depth feature extraction based on CNN and similarity calculation. Feature extraction consists of two parallel CNNs whose structure are same, including 4 convolutional layers, 4 pooling layers and a fully connected layer. Each CNN respectively processes the fused input image to obtain its feature maps. At the stage of similarity calculation, compute the similarity between the feature vectors output by two CNNs and determine if they belong to the same person. Obviously, the purpose of training twin neural network is to make the distance between the image features belonging to one person keeps the closest.

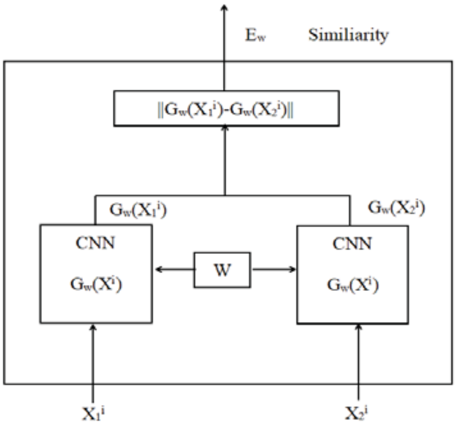


Figure 4. Structure of twin neural network structure.

Let X_1^i and X_2^i be the two samples in the i^{th} pair, $G_w(X_1^i)$ and $G_w(X_2^i)$ be the network outputs, $y^i \in \{0,1\}$ be the binary label to indicate X_1^i matches X_2^i or not. If they are similar, then $y^i = 1$. Otherwise $y^i = 0$. The cost function we choose to measure a sample pair (X_1^i, X_2^i) is shown in (8}.

$$L(w^i, y^i, X_1^i, X_2^i) = -(y^i \log\frac{1}{1+e^{D_w}} + (1-y^i)\log(1-\frac{1}{1+e^{D_w}})) \quad (8)$$

where D_w is the Euclidean distance between the outputs of twin neural network, shown in (9).

$$D_w(X_1^i, X_2^i) = \| G_w(X_1^i) - G_w(X_2^i) \| \quad (9)$$

For all training samples, the total loss function $L(w)$ is expressed in (10).

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} L(w^i, y^i, X_1^i, X_2^i) \quad (10)$$

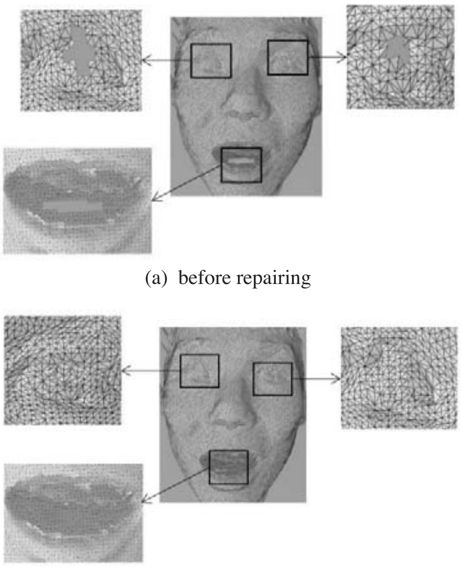
V. Experiments and Analysis

To test the performance of our method, we employ CASIA-3D which includes 4,624 face images of 123 people in a laboratory environment. Each subset belonging to a person contains 37 to 38 cloud data of frontal faces (tilted degree no more than 20 degrees) by different poses, expressions and lighting. To train and test the network performance, we generate new point clouds and their corresponding RGB images by rotation.

A. Effect of Hole Repairing

Fig. 5 shows the visual comparison between non-repairing and repairing.

It can be easily observed from Fig. 5 that there exist local holes in the original point cloud, especially in the eyes and mouth. After repairing, new Delaunay triangle patches are generated in the local holes. And the new point cloud has better integrity compared with the original 3D face model.



(a) before repairing

(b) after repairing

Figure 5. Visual Model generated by non-repairing or rrepairing.

B. *Effect of Feature Fusion*

To test the effect of feature fusion, we randomly divided the faces into four groups for training, validation and testing. Each group includes 30, 30, 30 or 33 persons. In the iteration train stage by gradient descent, we train the network 2000 times. And the number of positive image pairs or negative pairs is 10,000. In the test stage, we randomly choose 100 point clouds with their corresponding RGB images to test.

Table 1 shows the recognition rates on the test set. 2D or 2D+3D indicates that only the RGB image or the RGB image with depth map is used as the input source of CNN.

TABLE I. THE COMPARISONS OF RECOGNITION ACCURACY ON FEATURE FUSION OR NOT

	2D	**2D+3D**
1	83.73%	95.24%
2	84.34%	92.65%
3	88.52%	97.10%
4	86.76%	94.83%

Table 1 shows that the recognition accuracy using feature fusion is much higher than just using 2D texture from only RGB image. Especially for the third group, the accuracy increases about 6%. The main reason for significant improvement is by combining 3D depth map, our method can greatly decrease the influence of posture, light and other factors.

C. *The Influence of Training Scale*

In order to verify that the performance of CNN-based twin neuron networks is less affected by the number of training samples, a set of experiments are carried on using 1520, 1000, and 500 training samples on classic CNN, CNN+ and our method by CNN-based twin neuron networks. Here, CNN means classic CNN with only 2D image texture, CNN+ means classic CNN with 2D texture and depth map. Table 2 gives the comparison results.

TABLE II. RECOGNITION ACCURACY OF DIFFERENT NUMBERS OF TRAINING SAMPLES

	500	**1000**	**1520**
Classic CNN	79.57%	90.63%	95.25%
CNN+	83.82%	95.19%	98.81%
Our Method	94.23%	97.52%	99.15%

Table 2 shows that the recognition accuracy of our method remains the smallest changes than that of the other two ways. The recognition rate of our method can be up to of 94% with only 500 training samples, which is very close to the recognition rate of CNN+ using 1000 training samples.

VI. CONCLUSIONS

Aiming at improving the robustness of 3D face recognition, we design a method based on deep twin neural network. This method combines RGB 2D face with depth map reflecting facial 3D information as the input source of each CNN. The experimental results show that our method can achieve better performance than traditional CNN frame, including in the case of small training set. In the future, we can improve recognition ability from loss function, back propagation algorithm and gradient descent algorithm.

ACKNOWLEDGMENT

This research is partially sponsored by the National Key R&D Program of China under Grant No. 2017YFB1302203.

REFERENCES

[1] Di Tang.Zhe Zhou, and Yinqian Zhang, "Face Flashing: a Secure Liveness Detection Protocol based on Light Reflection,". 2018.

[2] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun, "Blessing of Dimensionality: High-dimensional Feature and Its Efficient Compression for Face Verification," CVPR 2013

[3] Xu C,Wang Y, Tan T, et al. "Automatic 3D face recognition combining global geometric features with local shape variation information," IEEE International Conference on Automatic Face and Gesture Recognition, 2016. pp. 308-313.

[4] Chua C S, Han F, and Ho Y K, "3D Human Face Recognition Using Point Signature," IEEE International Conference on Automatic Face and Gesture Recognition, 2010. pp. 2010: 233.

[5] Zhong C, Sun Z, and Tan T, "Robust 3D Face Recognition Using Learned Visual Codebook," IEEE Conference on Computer Vision & Pattern Recognition. 2014, pp. 1-6.

[6] Syed Zulqarnain Gilani and Ajmal Mian, "Learning from Millions of 3D Scans for Large-scale 3D Face Recognition," IEEE Conference on Computer Vision and Pattern Recognition, 2018.