

Article

Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion

Jing Li ^{1,2,†}, Tao Qiu ^{3,†}, Chang Wen ^{3,*}, Kai Xie ^{1,2} and Fang-Qing Wen ^{1,2}

¹ School of Electronic and Information, Yangtze University, Jingzhou 434023, China;

201501479@yangtzeu.edu.cn (J.L.); 500646@yangtzeu.edu.cn (K.X.); wenfangqing@yangtzeu.edu.cn (F-Q.W.)

² National Demonstration Center for Experimental Electrical and Electronic Education, Yangtze University, Jingzhou 434023, China

³ School of Computer Science, Yangtze University, Jingzhou 434023, China; 201603441@yangtzeu.edu.cn

* Correspondence: 400100@yangtzeu.edu.cn; Tel.: +86-136-9731-5482

† These authors contributed equally to this work.

Received: 20 May 2018; Accepted: 25 June 2018; Published: 28 June 2018



Abstract: Given that facial features contain a wide range of identification information and cannot be completely represented by a single feature, the fusion of multiple features is particularly significant for achieving a robust face recognition performance, especially when there is a big difference between the test sets and the training sets. This has been proven in both traditional and deep learning approaches. In this work, we proposed a novel method named C2D-CNN (color 2-dimensional principal component analysis (2DPCA)-convolutional neural network). C2D-CNN combines the features learnt from the original pixels with the image representation learnt by CNN, and then makes decision-level fusion, which can significantly improve the performance of face recognition. Furthermore, a new CNN model is proposed: firstly, we introduce a normalization layer in CNN to speed up the network convergence and shorten the training time. Secondly, the layered activation function is introduced to make the activation function adaptive to the normalized data. Finally, probabilistic max-pooling is applied so that the feature information is preserved to the maximum extent while maintaining feature invariance. Experimental results show that compared with the state-of-the-art method, our method shows better performance and solves low recognition accuracy caused by the difference between test and training datasets.

Keywords: face recognition; convolutional neural network; color 2-dimensional principal component analysis; decision-level fusion; normalization; layered activation function; probabilistic max-pooling

1. Introduction

Biometric recognition technology is a method for identification or verification based on human physiological or behavioral characteristics. Biometric features for authentication include iris, fingerprint, palm print, face, ear type, voiceprint and gait. Face recognition is face-based identification technology, which is a main trend of biometric recognition. Compared with other biometric technologies, face recognition technology has unique advantages. It does not require user cooperation and the acquisition process is non-contact. Meanwhile, it has better concealment. The benefits make it particularly suitable for security monitoring and airport anti-terrorism applications.

Face recognition is generally divided into three basic technical lines [1]. The first type is 2D face recognition based on pictures. 2D face recognition started earlier and has achieved great success. Before 2003, most of the work about face recognition focused on 2D face recognition, which mainly included: (1) recognition algorithms based on geometric features [2,3]; (2) recognition algorithms based on subspace [4,5]; (3) recognition algorithms based on elastic matching [6,7]; (4) recognition

algorithms based on neural networks [8,9]. 2D face recognition identified a face with the brightness image, which is inevitably influenced by factors such as illumination and which is also the main bottleneck for the further development of 2D face recognition.

The second category is 3D face recognition based on 3D geometric information. In recent years, attention increasingly has been placed on 3D face recognition. Liang proposed a new method to handle pose variations in 3D face recognition [10] and Abbad makes a great contribution to the problem of expression variations in 3D face recognition [11]. Apparently, the remarkable advantage of 3D data over 2D data is that it has the original geometric information of the human face. Two-dimensional face images are easily affected by factors such as lighting, posture, and occlusion, which will lead to a decrease in the recognition ability of the face recognition system. However, a 3D face recognition algorithm is hardly affected by these factors. There is more potential in solving face problems with variable illumination and posture. This is because the three-dimensional data can truly reflect the surface geometric features of the human face and will not lose information due to changes in posture. Moreover, if the acquisition process of three-dimensional data is not considered, since 3D data has no brightness information, the three-dimensional data will not be affected by the change of illumination. Owing to these, 3D face recognition is booming. Dagnes reviews a series of 3D face occlusion recognition methods [12]. Facial occlusions are general situations which is hard to address in 2D face recognition, however, it has been preferably solved in 3D face recognition. Vezzetti proposes a novel automatic method for facial landmark localization relying on geometrical properties of 3D facial surface in the presence of occlusions [13]. In other fields, 3D recognition technique can also be applied. Moos proposed a methodology to automatically diagnose and formalize prenatal cleft lip [14]. Similarly, in the literature [15], Moeini proposed a feature extraction to handle the problem of facial appearance changes including facial makeup and plastic surgery in face recognition. Meanwhile, other scholars make distinctive contributions to 3D face recognition [16–18].

The third type of face recognition technology, namely the 2D+3D dual-mode fusion technique [19,20], has achieved good performance. The 2D+3D effectively combines the more mature 2D image recognition technology with the three-dimensional shape attributes of the face to obtain a better face recognition performance. In recent years, there has been progress in 3D-2D face recognition [21,22].

Deep learning has grown rapidly on account of the emergence of large-scale face datasets, whose powerful data learning has brought a research upsurge to 2D face recognition. It can effectively solve the problems of traditional machine learning algorithms. Our work focuses on the application of state-of-the-art deep learning technology in face recognition.

Deep learning [23] is a hot area in machine learning research by building deep neural networks to simulate the mechanism of human brain and then interpreting and analyzing data such as image, voice and text [24,25]. The effectiveness of traditional machine learning depends largely on the performance of handcrafted feature representation. During this process, the role of machine learning approaches is only to optimize learning weights and ultimately to produce optimal learning outcomes [26]. Different from the traditional machine learning methods, deep learning tries to automatically complete the work of data representation and feature extraction [27,28]. The main power of a convolutional neural network (CNN) lies in the deep architecture, which allows for extracting a set of discriminating feature representations at multiple levels of abstraction. In recent years, CNN has been widely applied in face recognition because of its good performance [29,30]. The success of CNN is attributed to its ability to learn rich image features. However, training a deep CNN network relies on learning millions of network parameters and requires plenty of labeled datasets for pre-training [31].

In practice, we do not usually train an entire CNN network from scratch, since it is impossible to collect a dataset with sufficient size to meet the needs of the CNN for each new face recognition task [32]. As an alternative, it is common to pre-train the CNN model on a large dataset and then use the weights of this model as an initialization or as a specific feature extractor for the task of interest, then finally fine tune the network model using the test set [33,34].

However, when the test and training datasets have great differences in illumination, expression, viewpoints or other factors, even the fine-tuned CNN cannot achieve a good recognition performance. The fundamental cause is that fine tuning is still based on the pre-trained CNN model which limits the recognition performance for the current recognition task. Effective fine tuning requires that all layers of the pre-trained model need to be fine-tuned when the variations between the source and target applications are significant. However, the limited fine-tuned data can cause over-fitting when the pre-trained CNN is fine-tuned too much [35]. CNN completes the classification and recognition task mainly through the study of the distribution of data [36,37]. When there is large difference between test and training set, the initial data distribution obtained by the pre-trained CNN network is difficult to change, which makes it difficult to improve the adaptability of the CNN model even if the CNN model is fine-tuned.

Another problem of the training network is the large number of network parameters and the fact that it is time consuming. The reason is that the parameters' change of any layer will cause the change of the distributions for input data in subsequent layers, which leads to the neural network constantly needing to adapt to the new data distribution. Thus, tuning parameters carefully and training with smaller learning rate are particularly important. Moreover, non-linear saturation of the activation operation makes it more difficult [38,39]. LeCun proposed that data normalization might speed up network convergence [40]. In 2015, batch normalization was proposed to solve the internal covariate shift in [41].

Aiming at the above problems, we propose a method of face recognition using a color 2-dimensional principal component analysis-convolutional neural network (C2D-CNN) to make full use of color information and complementary information for face recognition.

For color information, there have been some scholars [42–44] who have noticed its importance to face recognition. Color face recognition is usually conducted on each color channel and the final result comes from the fusion of the three color channels. Similarly, CNN networks make full use of three color channels of RGB for feature extraction, respectively. All of them omit the inner correlations among color channels. A color channel fusion method using a joint dimensionality reduction algorithm was proposed by [45] which can effectively improve the performance of color face recognition. In contrast, color 2-dimensional principal component analysis (2DPCA) is based on the concept of color value including three channels, which devotes to making full use of color information and the correlation between color channels. Color 2DPCA is devised to combine color and spatial information. It still has good robustness for the test and training set with large differences. So, color 2DPCA provides an effective way to improve the robustness of CNN.

The contributions of this paper are summarized as follows:

- We propose a deep face recognition method, C2D-CNN, which combines the two features into the decision-making level, with high-accuracy and low computational cost.
- We investigate a new CNN model. Through careful design, (1) normalization is introduced to accelerate the network convergence and shorten the network pre-training time; (2) a layered activation algorithm is added to improve the non-linear function of the activation function and solve the problem of gradient saturation and gradient diffusion; (3) probabilistic max-pooling is applied to preserve the feature in maximum extent while maintaining feature invariance.

The remainder of this paper is organized as follows: Section 2 presents the C2D-CNN model; Section 3, the experimental results are analyzed and discussed; the conclusion is given in Section 4.

2. Face Recognition with Color 2-Dimensional Principal Component Analysis-Convolutional Neural Network (C2D-CNN) Model

2.1. Overview of the Proposed Method

The face recognition method using a C2D-CNN model under decision-level fusion is divided into three parts: feature extraction for CNN, feature extraction for color 2DPCA, and decision-level fusion. The algorithm flow is shown in Figure 1.

Figure 1 shows the overall flow of the proposed method, C2D-CNN. In the proposed algorithm, firstly, the input image is scale normalized. Then, our algorithm is divided into three parts, including CNN feature extraction, color 2DPCA feature extraction and decision-level fusion. Furthermore, CNN and color 2DPCA are respectively applied to extract features from the normalized images. Among them, the CNN model is trained on the FRGC (Face Recognition Grand Challenge) v2.0 dataset to obtain the depth features. Color 2DPCA extracts handcrafted features that contain richer color information and spatial information. In order to fuse the feature vectors, Mahalanobis distance and Euclidean distance are respectively used to calculate the similarity of the feature vectors. Finally, the similarity weight ratio is set (the setting method will be described in detail in the Section 3.3.3) to fuse the feature vectors in decision level.

2.2. Feature Extraction with CNN

The CNN model proposed in this paper consists of convolution layer, normalization layer, layered activation function layer, probabilistic max-pooling layer, and fully connected layer. In this algorithm, CNN is first pre-trained with large samples, and then the pre-trained CNN model (Softmax classifier is removed) is used as feature extractor to extract facial features. The structure of the CNN network is shown in Figure 2.

The input of the convolutional neural network are color face images which have been scale normalized. Firstly, the input image is convoluted through several different convolution kernels to manipulate and extract the image features. The filter coefficients are obtained automatically during the training process and are dependent on the characteristics of the input training images. Then, the convolutional operation results are normalized in the normalization layer, which effectively prevents the network from over-fitting and ensures effective gradient descent. In order to increase the non-linear expression ability of the network, the normalized data is input into the layered activation function layer. Finally, the result of the activation function is pooled, retaining the most significant features and improving the model's distortion tolerance. After several convolutional layers and pooled layers, the acquired salient features are processed through the fully connected layer to obtain CNN feature representation results.

The pre-training of the CNN mainly includes the process of forward learning network parameters and back propagation of network parameters.

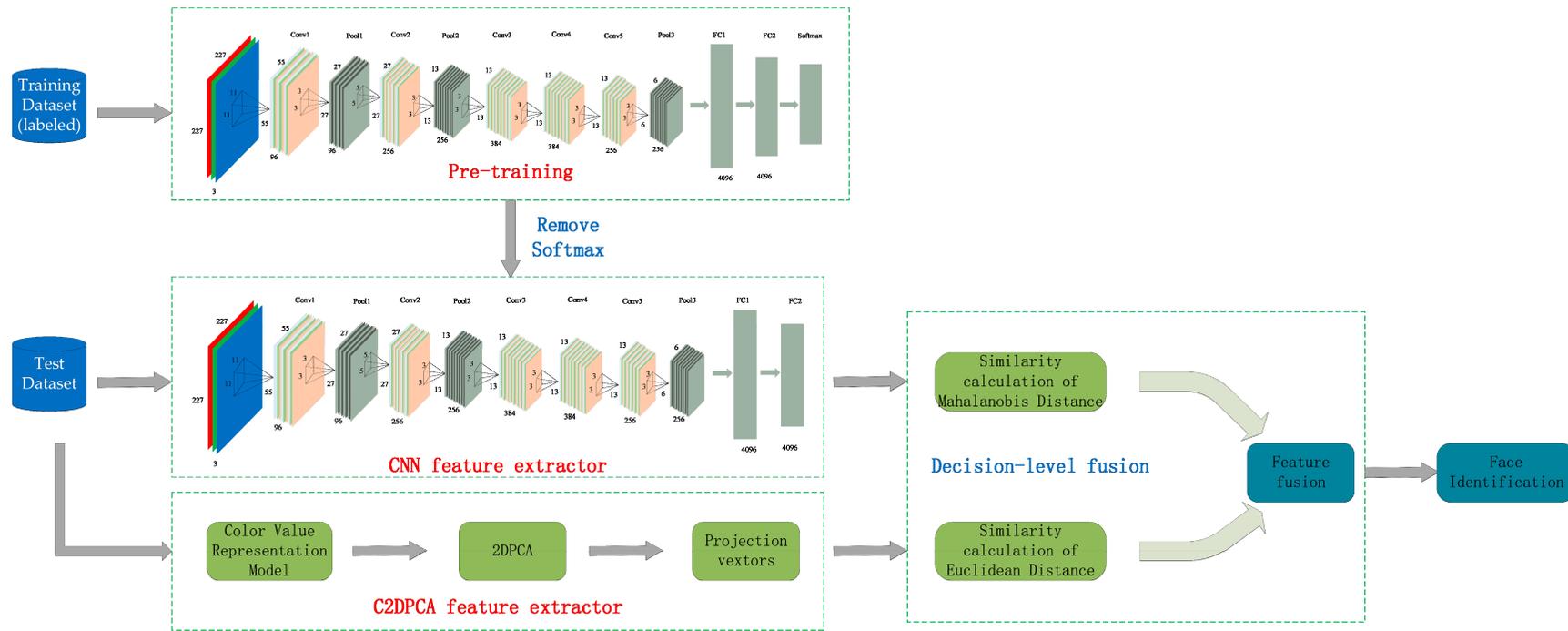


Figure 1. The flow diagram for face recognition using color 2-dimensional principal component analysis-convolutional neural network (C2D-CNN).

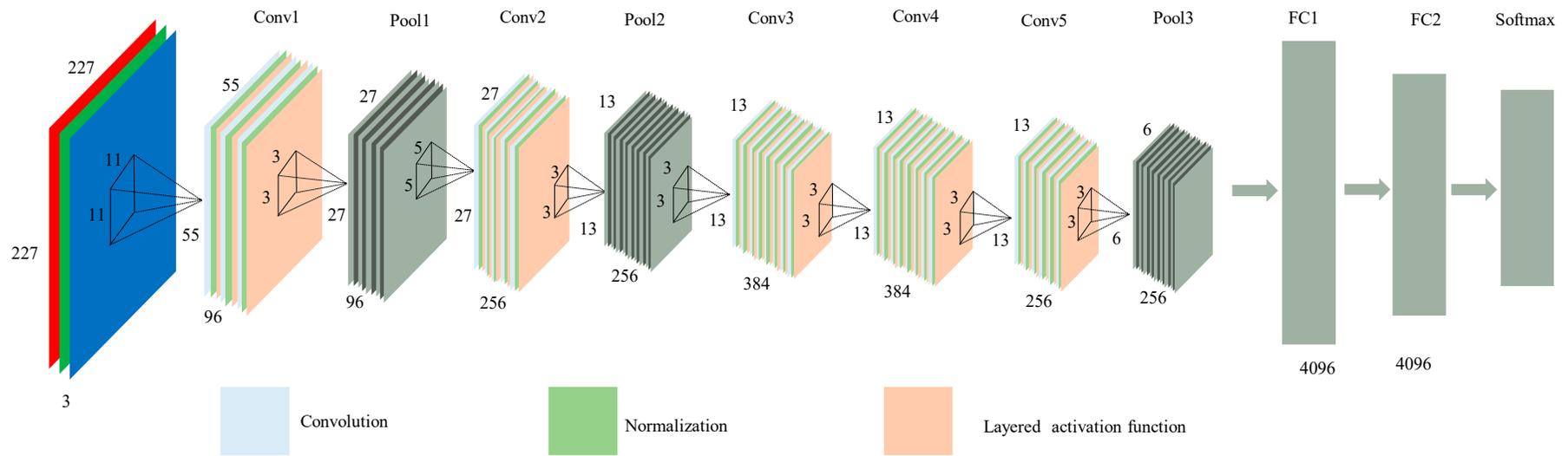


Figure 2. The structure of the CNN network based on the face recognition.

2.2.1. Forward Propagation of CNN Network

The parameters of the network were learned through the forward propagation. In each layer, the input features are calculated as follows:

$$k^{(l+1)} = LA(BL(F^{(l+1)}k^{(l)} + b^{(l+1)})) \quad (1)$$

$k^{(l)}$ is the vector output of the l -th layer, $k^{(l+1)}$ is the vector output of the $(l + 1)$ -th layer, $F^{(l+1)}$ is a matrix of the inter-layer linear coefficients, $b^{(l+1)}$ is the vector composed of biases, $LA(\cdot)$ is the layered activation function, and $BL(\cdot)$ is the normalization algorithm proposed in this paper.

(1) Normalization algorithm

In the training process of the CNN, due to the changes of network parameters, the distribution of input data in each layer might change, which in turn leads to different learning rates required for different dimensions. While pre-training the network, it is usually necessary to select the minimum learning rate, thus preventing the network from over-fitting and ensuring the effective decline of the gradient. To address this problem, we use normalization to process the data of each dimension and make it satisfy the same distribution. Normalization can prevent the occurrence of over-fitting when still using a large learning rate.

Supposing that the output of a layer contained η dimensions, $x = \{x^{(1)}, x^{(2)}, \dots, x^{(\eta)}\}$, the normalization was applied independently to each dimension. Then, we take any dimension as an example for explaining. In a mini-batch sample with the size of s , the data of the mini-batch can be defined as: $B_x = x_1, x_2, \dots, x_s$. The normalized sample data is B_y , $B_y = y_1, y_2, \dots, y_s$, y_i ($i \in [1, s]$) obeys the standard normal distribution with zero mean and unit variance.

In Algorithm 1, the initial value of μ_B was set as 0 and the initial value of σ_B^2 was set as 1. γ is the learning rate related to the momentum values ϕ and ψ , having been set as 0.01 by default. The specific combination coefficient ϕ and ψ were learned from the data, thus ϕ and ψ are self-adaptive to the input data. During the test phase, μ_B and σ_B^2 took on the final training value.

Algorithm 1 Normalization

Input: CNN Network and mini-batch B_x

Output: Normalized sample data B_y

1. Mini-batch mean: $\mu = \frac{1}{s} \sum_{i=1}^s x_i$
 2. Mini-batch variance: $\sigma^2 = \frac{1}{s} \sum_{i=1}^s (x_i - \mu)^2$
 3. Normalized value: $y_i = \frac{x_i - \mu}{\sigma}$
 4. Update the global mean: $\mu_B = (1 - \phi) * \mu_B + \phi * \mu$
 5. Update the global variance: $\sigma_B^2 = (1 - \psi) * \sigma_B^2 + \psi * \sigma^2$
 6. Update the momentum value ϕ : $\phi = \phi - \gamma \frac{\partial L}{\partial \phi}$
 7. Update the momentum value ψ : $\psi = \psi - \gamma \frac{\partial L}{\partial \psi}$
-

During the training phase, the back-propagation gradient of the normalization layer was determined by the chain rule:

$$\frac{\partial L}{\partial \sigma^2} = -\frac{1}{2} \sum_{i=1}^s \frac{\partial L}{\partial y_i} (x_i - \mu) (\sigma^2)^{\frac{3}{2}} \quad (2)$$

$$\frac{\partial L}{\partial \mu} = \left(\sum_{i=1}^s \frac{\partial L}{\partial y_i} \frac{-1}{\sigma} \right) + \frac{\partial L}{\partial \sigma^2} \frac{-2 \sum_{i=1}^s (x_i - \mu)}{s} \quad (3)$$

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y_i} \frac{1}{\sigma} + \frac{\partial L}{\partial \sigma^2} \frac{2(x_i - \mu)}{s} + \frac{1}{s} \frac{\partial L}{\partial \mu} \quad (4)$$

Given the output loss function L , we can learn the combination coefficient ϕ and ψ . The back propagation for this learning is given by:

$$\frac{\partial L}{\partial \phi} = \frac{\partial L}{\partial \mu_B} \frac{\partial \mu_B}{\partial \phi} = \frac{\partial L}{\partial \mu_B} (\mu - \mu_B) \quad (5)$$

$$\frac{\partial L}{\partial \psi} = \frac{\partial L}{\partial (\sigma_B^2)} \frac{\partial (\sigma_B^2)}{\partial \psi} = \frac{\partial L}{\partial (\sigma_B^2)} (\sigma^2 - \sigma_B^2) \quad (6)$$

Due to the parallelism of the calculation, the normalization in the mini-batch was more effective than in all dimensions, so the data was normalized by using mini-batch to satisfy the same distribution. In addition, ideally, the mean and variance should be specific to all dimensions, which was impractical. Therefore, the mean and variance of all dimensions was estimated using the mean and variance of the mini-batch data, with it all and partially linked by updating the global variance and mean, thus simplifying the normalized calculation process. In the experiment, it was found that the recovery operation in BN had little effect on improving the speed of the network training. To further improve the convergence rate, the normalization layer removed the recovery operation, which worked better than BN.

The normalization on the mini-batch data alleviated the problem of Internal Covariate Shift and non-linear saturation, which greatly reduced the training time of the network and effectively accelerated the convergence of the network.

(2) Layered activation function

The purpose of the activation operation is to introduce the non-linearity into the CNN network. Furthermore, we introduce the layered activation function, which might address 3 problems: (1) adaptivity is introduced to the activation function; (2) improve the non-linear expression ability; (3) solve the problems such as gradient saturation, gradient diffusion. The structure of the layered activation function is shown in Figure 3.

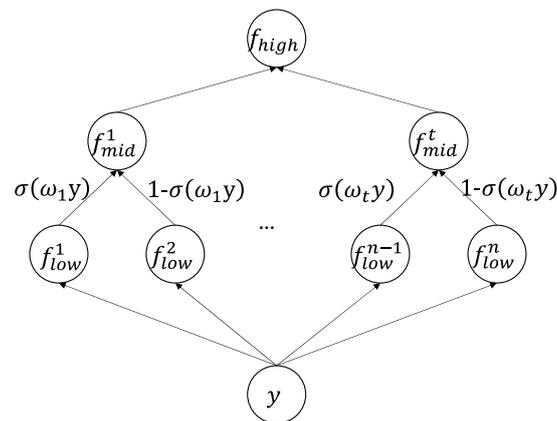


Figure 3. The structure of the layered activation function.

Figure 3 shows a three-level structure in which low-level nodes are associated with learnable basic activation functions. Each middle-level node corresponds to a combined output of a pair of low-level nodes in a binary tree structure, and a high-level node corresponds to the overall output. Among them, y is the normalized input parameter, and $f_{low}, f_{mid}, f_{high}$ denote low-level nodes, middle-level nodes, and high-level node functions, respectively. The three-level node function is given by:

$$f(y) = \begin{cases} f_{low}^n(y), & \text{low - nodes} \\ \sigma(\omega y)f_{mid}^{k,left}(y) + (1 - \sigma(\omega y))f_{mid}^{k,right}(y), & \text{middle - nodes} \\ \max_{1 \leq k \leq t} f_{mid}^k(y), & \text{high - nodes} \end{cases} \quad (7)$$

where, $f_{low}^n(\cdot)$ is represents the n -th low-level node function ($f_{prelu}(\cdot)$ or $f_{elu}(\cdot)$), $\sigma(\cdot)$ is sigmoid function, $f_{mid}^{k,left}(\cdot)$ and $f_{mid}^{k,right}(\cdot)$ are the left or right child node functions of the k -th middle node respectively. The basic activation function is given by:

$$f_{prelu}(y) = \begin{cases} y, & y > 0 \\ \alpha y, & y \leq 0 \end{cases} \quad (8)$$

$$f_{elu}(y) = \begin{cases} y, & y > 0 \\ \beta(e^y - 1), & y \leq 0 \end{cases} \quad (9)$$

In the formula, α and β are the weights of the negative part of control, which are learnt from data. The initial value of α is set as 0.25 and β is set as 1.0 [46]. $f_{mid}^k(\cdot)$ represents the k -th middle-node function, given by:

$$f_{mid}(y) = \sigma(\omega y)f_{prelu}(y) + (1 - \sigma(\omega y))f_{elu}(y) \quad (10)$$

The high-level node function $f_{high}(\cdot)$ is the maximum of t middle-nodes function.

The advantages of the two basic activation functions are: The $f_{prelu}(\cdot)$ allows the slope of the negative part to be learned from the data, which improves the expression ability of the activation function; the linear component of $f_{elu}(\cdot)$ makes it possible to mitigate the gradient disappearance and the soft-saturation can reduce the slope from a predefined or fixed threshold to zero, which can accelerate model learning. The basic activation function is shown in Figure 4.

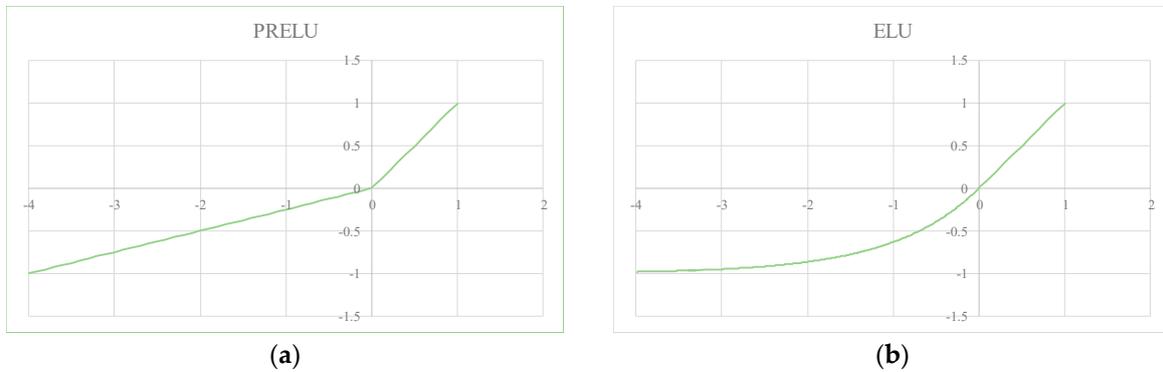


Figure 4. Activation functions of low-nodes: (a) PRELU; (b) ELU.

The middle-level node function $f_{mid}(\cdot)$ can adaptively choose the combination coefficient of the two basic activation functions. Specifically, we instead learn a gating mask ω , rather than directly learning a combination coefficient. The output of the gating mask and y is jointly fed to a sigmoid function to generate the combination coefficient.

The output of the layered activation function $f_{high} = \max_{1 \leq k \leq t} f_{mid}^k(y)$ is the maximum of t middle-level nodes. The competition mechanism is introduced to enhance the non-linearity of the activation operation. Only the activation neuron with the highest activation can be activated while other neurons were inhibited.

The algorithm learns the layered activation function in a data-driven manner. $f_{prelu}(\cdot)$ and $f_{elu}(\cdot)$ functions are introduced to solve the gradient diffusion problem. By adaptively combining them,

the layered activation function has a good adaptability. The competition mechanism is introduced to improve the ability to learn non-linear transformation.

(3) Probabilistic max-pooling

Next is the pooling layer, whose main task is to blur the features so as to obtain the invariance of shift and scale. The state of neurons in this layer is determined by the upper local “accepted domain” and the pooling rules of the upper layer. In general, the minimum, mean, median and maximum of the neuron response values in the “accepted domain” are taken as the pooling rules. Although these rules can achieve the above requirements, however, they are irreversible. In the process of back propagation, the loss of feature representations is inevitable, which directly restricts the robustness of CNN. A probabilistic max-pooling approach [47] is applied so that the feature information is preserved to maximum extent while maintaining feature invariance: when there is, as long as a neuron is opened, the response will be generated by the sample value S ; otherwise, S does not respond, which can be expressed as:

$$P(S = 1|k) = \sum_{i,j \in C} \exp((F^l * k)_{ij} + b_l) / (1 + \sum_{i,j \in C} \exp((F^l * k)_{ij} + b_l)) \quad (11)$$

2.2.2. Back Propagation (BP) of CNN Networks

To improve the self-adaptability of the new network, the back propagation (BP) algorithm [48] was used to adjust the parameters in reverse. In view of the fact that the parameters of the network update very slowly by using the variance loss function, the cross-entropy cost function [49] was adopted, whose advantage lies in that large errors can lead to quick updates of the network’s parameters while small errors can slowly update the network’s parameters. As for the voiceprint training sample set with the size of N , $e = \{(e^{(1)}, v^{(1)}), \dots, (e^{(N)}, v^{(N)})\}$, the cross entropy cost function was defined as:

$$\Gamma(\theta) = -\frac{1}{N} \sum_{i=1}^N [v^{(i)} \log(o^{(i)}) + (1 - v^{(i)}) \log(1 - o^{(i)})] \quad (12)$$

where $o^{(i)}$ denotes the actual output corresponding to the input $e^{(i)}$, and $v^{(i)}$ denotes the category tag corresponding to the i -th group of data, $v^{(i)} \in \{1, 2, 3, \dots, k\}$, k is the number of face categories, and the back-propagation gradient of the convolutional parameters w and b were determined by the following formulas:

$$\frac{\partial}{\partial w^{(i)}} \Gamma(\theta) = \frac{1}{N} \sum_{i=1}^N x^{(i)} (o^{(i)} - v^{(i)}) \quad (13)$$

$$\frac{\partial}{\partial b^{(i)}} \Gamma(\theta) = \frac{1}{N} \sum_{i=1}^N (o^{(i)} - v^{(i)}) \quad (14)$$

The calculation formulas of the updated parameters $w^{(i)}$ and $b^{(i)}$ are as follows:

$$w_i^l = w_i^l - \rho \frac{\partial}{\partial w_i^l} E^\tau \quad (15)$$

$$b_i^l = b_i^l - \rho \frac{\partial}{\partial b_i^l} E^\tau \quad (16)$$

where, ρ is the learning rate, and E^τ is the error of training face samples for the current batch face images.

2.2.3. Feature Extraction

In this algorithm, a pre-trained CNN model is used as a feature extractor to extract the facial features. The 4096-dimensional output of the last fully connected layer is used as the eigenvector extracted by CNN. The structure of the fully connected layer is shown in Figure 5.

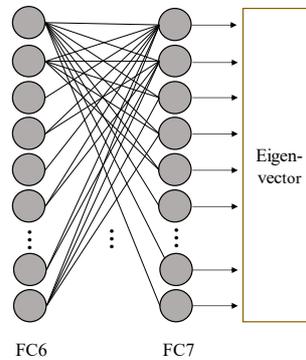


Figure 5. The structure of the fully connected layer.

The output of the fully connected layer is determined by the following formula:

$$\rho^{(l+1)} = \psi(W^{(l+1)}\rho^{(l)} + b^{(l+1)}) \quad (17)$$

$\rho^{(l)}$ is the output vector of the l -th layer, $\rho^{(l+1)}$ is the output vector of the $(l + 1)$ -th layer, $W^{(l+1)}$ is a matrix of the inter-layer linear coefficients, $b^{(l+1)}$ is the vector composed of biases, and $\psi(\cdot)$ is the activation function.

2.3. Feature Extraction with Color 2-Dimensional Principal Component Analysis (2DPCA)

Traditional 2DPCA [50] and PCA [51] need to convert the original image to a grayscale version before extracting the feature, so the color information is usually ignored. Color 2DPCA introduces the concept of color value (taking R, G, B color components as a whole) and applies the existing 2DPCA framework to color face recognition directly [52]. Compared with most of the previous methods which make use of three color channels respectively, color 2DPCA can obtain more color information and spatial information.

First, the composite vector representation of the color face image A is defined as:

$$A = (c_1, c_2, \dots, c_n)^T \quad (18)$$

Among them, the basic element c_i is denoted as the i -th color value (the color vector of the pixel). Once the basic operations are defined for color values, the color values can be regarded as normal scalar values for computing, now we define the basic operations of the color value: (1) addition: the sum of the color values is the sum of the two color vectors; (2) subtraction: subtraction of the color value is the difference between the two color vectors; (3) multiplication: the product of the two color values are the inner product of the two color vectors. In this algorithm, multiplication is mainly used to calculate the covariance of two features, reflecting the correlation of two color values. If the result of multiplication is zero, they are independent, otherwise they are positive or negative correlation.

Then, the composite vector A of the image is projected onto the vector X , the projection feature vector Y of A can be obtained as follows:

$$Y = AX \quad (19)$$

Suppose each person has M samples, with \bar{A} representing the average value of M composite vectors, the sample matrix is defined as:

$$D = [(A_1 - \bar{A})^T, (A_2 - \bar{A})^T, \dots, (A_M - \bar{A})^T]^T \quad (20)$$

where D is a composite matrix (the basic elements of D are color value rather than scalar values). Each element in row represent a centered sample and T represents the transpose. According to PCA, the optimal projection vector can be obtained by calculating $D^T D$.

$$D^T D = \sum_j^M (A_j - \bar{A})^T (A_j - \bar{A}) = M \times G_t \quad (21)$$

Among them, the sample covariance matrix G_t is defined as:

$$G_t = \frac{1}{M} \sum_{j=1}^M (A_j - \bar{A})^T (A_j - \bar{A}) \quad (21)$$

In this way, 2DPCA can be applied to find the optimal projection of the image composite vector A . The total scatter $J(X)$ of the projection sample is introduced to detect the discriminative ability of the projection vector. $J(X)$ can be represented by the trace of the covariance matrix of the projection vector Y :

$$\begin{aligned} J(X) &= tr(S_X) \\ &= X^T E \left\{ (A - E(A))^T (A - E(A)) \right\} X \\ &= X^T G_t X \end{aligned} \quad (23)$$

The optimal projection axis $X_{opt} = \{X_1, X_2, \dots, X_d\}$ is the former d orthogonal column vectors.

$$\begin{cases} \{X_1, X_2, \dots, X_d\} = \operatorname{argmax} J(X) \\ X_i^T X_j = 0, i \neq j, i, j = 1, 2, \dots, d \end{cases} \quad (24)$$

Finally, the projection feature matrix of the image is obtained by using the best projection axis. The projection matrix of image A is:

$$Y^i = AX_{opt} = [AX_1, AX_2, \dots, AX_d] = [Y_1^i, Y_2^i, \dots, Y_d^i] \quad (25)$$

Y_i is the projection feature matrix extracted from the color 2DPCA.

2.4. Decision-Level Fusion

Because it is difficult to obtain good recognition performance when there is a big difference between test and training set, inspired by complementary theory, we combine color 2DPCA features with CNN features for face recognition. Color 2DPCA can extract rich color and spatial information. It has good robustness even when there is a big difference between test and training set. The Mahalanobis is widely used in face recognition for CNN [53,54], and the Euclidean distance is popular for PCA, 2DPCA and related algorithm [55,56]. They are respectively applied to the calculation of similarity. Finally, fusion of similarity weight is performed at the decision level.

For the CNN feature, suppose w_1'' and w_2'' are two learned eigenvectors for the gallery and query face images, respectively. The similarity G^{CNN} of the CNN eigenvector is calculated by the following formula:

$$G^{CNN} = \frac{(w_1'')^T \cdot w_2''}{|w_1''| \times |w_2''|} \quad (26)$$

where, $|\cdot|$ is the L2 norm, T is the transpose.

For the similarity of color 2DPCA features, the similarity between gallery image A_i and query image A_j is calculated as formula (27).

$$d(A_i, A_j) = \sum_{k=1}^d \|Y_k^i - Y_k^j\|^2 \quad (27)$$

where, $\sum_{k=1}^d \|Y_k^i - Y_k^j\|^2$ represents the Euclidean distance between two eigenvectors. Decision-level convergence is determined by the following equation:

$$G = \Omega G^{CNN} + (1 - \Omega) G^{C2D} \quad (28)$$

Among them, G is the fusion similarity, G^{C2D} is the normalized similarity of color 2DPCA, and Ω is the weight of similarity ($\Omega < 1$).

3. Experimental Results and Analysis

In this section, the experimental environment and the details of our paper will be described. All of the ideas of comparative experiments come from the proposed algorithm of this article, and they are applied to evaluate our method and demonstrate the correctness of our algorithm. The experiments are performed on the LFW (Labeled Faces in the Wild) and FRGC face databases and the results will be compared with the recognition rates in the relevant literature.

3.1. Dataset

In this paper, the LFW face database [57] and FRGC v2.0 face database [58] were selected for experiments to evaluate the face recognition algorithm proposed in this paper. The LFW database contains 13,233 face images collected from the Internet, among them, there are 1680 persons have two or more images and 4096 people have only one image. LFW is mainly used for research on face recognition under unrestricted conditions and has become the benchmark for academic evaluation of recognition performance. It can fully reflect the changes of face in real conditions, such as illumination, gesture, expression, background, gender etc. Figure 6 shows some of the pictures in the LFW database.



Figure 6. The software interface based on face recognition.

We report the averaged result over 10 runs of View2. The FRGC v2.0 face database includes 12,776 training images (including 6388 uncontrolled images and 6388 controlled images), 16,028 controlled object images and 8014 uncontrolled query images. The controlled images are of good quality, but the uncontrolled images are poor, such as illumination changes, low resolution and blurring, etc. FRGC Experiment 4 has been reported to be the most challenging FRGC experiment, so it is chosen to assess the face verification rate at the false accept rate of 0.1%. Figure 7 shows some of the pictures in the FRGC database.



Figure 7. The software interface based on face recognition.

In this paper, the recognition rate will be applied to evaluate the performance of the algorithm. N_r is the number of faces matched correctly, and N_a is the total number of face samples (including the number matching correct and mismatch faces). R is recognition rate, giving:

$$R = \frac{N_r}{N_a} \quad (29)$$

3.2. Experiment Setting

3.2.1. Training Details

The experiments were carried out in the operating system of ubuntu1604, with a NVIDIA GEFORCE GTX 1050 GPU, memory size of 16 GB, CPU as I7-7700, software platform on python 3.5, and tensorflow 1.2.1, the interface software for a cross-platform Qt machine. As is shown in Figure 8, software for face recognition was developed by using the proposed algorithm.

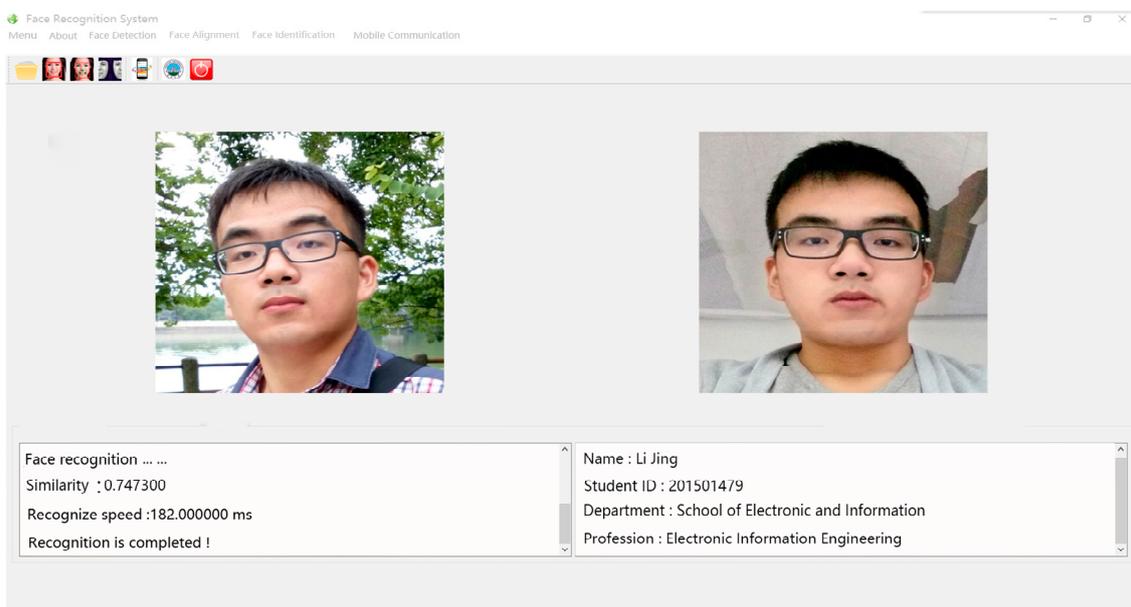


Figure 8. The software interface based on face recognition.

In this part, we will introduce the training and experiment details. In the following experiments, we will compare three models: CNN-0 (AlexNet), CNN-1 (AlexNet + BN), CNN-2 (Our model). In addition, we also report the accuracy under the LFW training and identification protocol. GTX 1050 GPU is applied to carry on parallel acceleration calculations. Other details about the pre-training network are shown in Table 1.

Table 1. The details of the CNN model.

	CNN-0	CNN-1	CNN-2
Learning rates	0.01	0.05	0.05
Dropout	0.5	—	—
Weight decay	0.001	0.001	0.001
Momentum	0.9	0.9	0.9
No. of epochs	80	25	20
Batch_size	512	512	512
Pooling method	Max-pooling	Max-pooling	Probabilistic max-pooling
Activation function	ReLU	ReLU	LAF
other	—	Batch Normalization	Normalization

In the following part, we will introduce the setting of the color 2DPCA. For the color 2DPCA, we need to determine its dimensions and the number of training and test samples, according to the actual experiment, when the dimension is greater than 30, its accuracy has tended to be stable and not increasing. We will select the training samples from two benchmark databases as the training set, and we know that the higher the proportion of training samples, the higher the accuracy. Therefore, we set the proportion of 1:3. The details about decision-level fusion are shown in the Table 2.

Table 2. The details of decision-level fusion.

	CNN-2	Color 2DPCA
Input	Color face images	Color face images
Output dimensions	4096	30
Matching strategy	Mahalanobis distance	Euclidean distance
Weight ratio	Depending on the sample database	

For the proposed algorithm C2D-CNN, there are a few other parameters that will be obtained in Section 3.3.

In the following experiment, when the similarity is greater than or equal to 0.7, the proposed algorithm considers face recognition to be successful. The following experimental data are the average result.

3.2.2. Data Augmentation Details

To train these three network models, 12776 FRGC database training images are selected as the training set. But the training set is too small to train a network model. To magnify the number of training samples, for each training sample of size $m \times m$ we extract all possible crops of size $(m - 4) \times (m - 4)$. Each crop is also flipped along its vertical axis yielding a total of $2 \times 5 \times 5 = 50$ kinds of crops. Then the crops are re-sized back to same size.

3.2.3. Fine-Tuning Details

Generally, there is a big difference between the dataset of the target task and the pre-trained dataset, regardless of the number of categories or image pattern. In the retrieval task of the target dataset, it is often difficult to achieve better performance by using the pre-trained CNN model to extract the visual features of the image. Therefore, in order to make the pre-trained CNN model parameters more suitable for the feature extraction of the target dataset, we use the image of the target dataset to fine-tune the pre-trained CNN model parameters. Fine-tuning training refers to retraining the CNN model on the FRGC training set. In the training process, only the output layer weights are initialized using Gaussian random numbers, and other weights of the CNN model are initialized using the corresponding weights of the pre-training model.

The difference from the pre-training is that a small learning rate should be set in the first 7 layers of the network, which can ensure that the parameters obtained through the pre-training CNN model are not destroyed during the fine-tuning process. For the last layer, setting a higher learning rate ensures that the entire network quickly converges on the target dataset to the new optimal point. For the CNN-0 network, in order to ensure the normal convergence of the network, the lr (learning rate) is reduced to 1/10 for every 20 epochs.

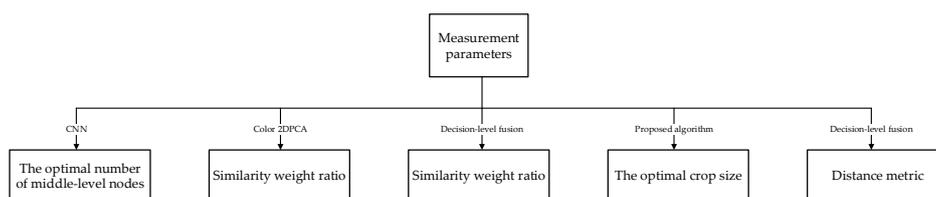
In Section 3.4.5, we will compare the effectiveness of fine-tuning and decision-level fusion. More details about the pre-training network are shown in the Table 1. The details about Fine-Tuning are shown in the Table 3.

Table 3. The details of Fine-Tuning.

	CNN-0	CNN-2
Learning rates	Conv1-FC2: 0.005 Softmax: 0.02	Conv1-FC2: 0.02 Softmax: 0.05
Dropout	0.5	—
Weight decay	0.001	0.001
Momentum	0.9	0.9
No. of epochs	60	20
Batch_size	512	512
Pooling method	Max-pooling	Probabilistic max-pooling
Activation function	ReLU	LAF
other	—	Normalization

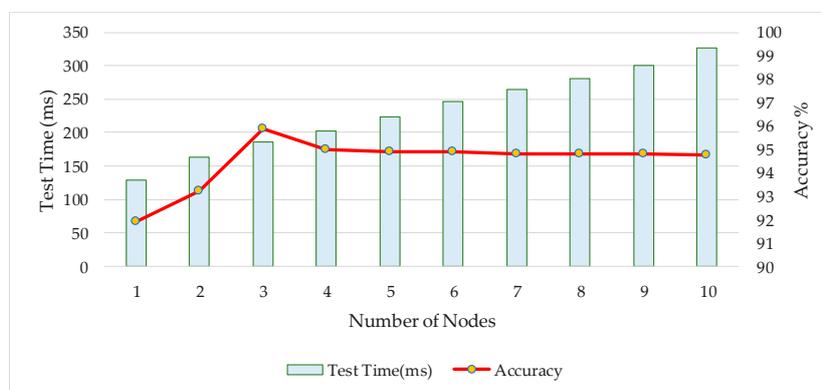
3.3. Acquisition of Testing Parameter

In this part, to give the proposed algorithm better robustness and recognition performance, firstly, we need to determine certain parameters of our proposed algorithm, and then we can safely apply it to specific recognition tasks. The parameters to be obtained are as follows: the number of layered activation function middle-nodes, the optimal feature dimension, fusion weight ratio, and image size. As the similarity weight ratio and feature dimension may vary with the benchmark face database used, two databases are used for experiments. For experiments unrelated to the face database, only the FRGC database will be applied. The specific flow is shown in Figure 9.

**Figure 9.** Acquisition of testing parameters.

3.3.1. Acquisition of the Optimal Number of Middle-Nodes in Layered Activation Function

In this experiment, we evaluated the influence of the number of the middle-level nodes on the recognition performance for the CNN network and expected to obtain the optimal number of nodes. The benchmark database FRGC was selected as the test dataset. Ten groups of experiments were carried out, the number of middle-level nodes is respectively: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. We evaluated the recognition rate and the averaged test time. The experimental results are shown in Figure 10.

**Figure 10.** Experiment of layered activation function middle-nodes setting.

As is shown in Figure 10, when the number of middle-level nodes is three, the optimal recognition rate is achieved. As the number of nodes increases, the recognition rate tends to be stable in a small area, however, the test time is significantly increased. In all the next experiments, the number of middle-level nodes will be set to three.

3.3.2. Acquisition of the Optimal Feature Dimension of Color 2DPCA in Different Benchmark Database

In this experiment, we compared the performance of the PCA-related algorithm under various feature dimensions and expected to obtain the optimal feature dimension of color 2DPCA. The benchmark database FRGC and LFW were selected as the test dataset. Ten groups of experiments were carried on. We compared 5 kinds of PCA related algorithm: PCA, CPCA (color PCA), 2DPCA (Two-Dimensional PCA), 2D2DPCA (2 Dimensional 2 Directional PCA), color 2DPCA. Among them, PCA is a basic dimension reduction algorithm, furthermore, CPCA can extract color feature, 2DPCA can extract spatial information and significantly reduce the dimensionality of features. The improvement of 2D2DPCA is that the transformation in ordinary 2DPCA only extracts the features in the rows of the data matrix, but the 2D2DPCA adds the transformation to the columns. The experimental results are shown in Figure 11.

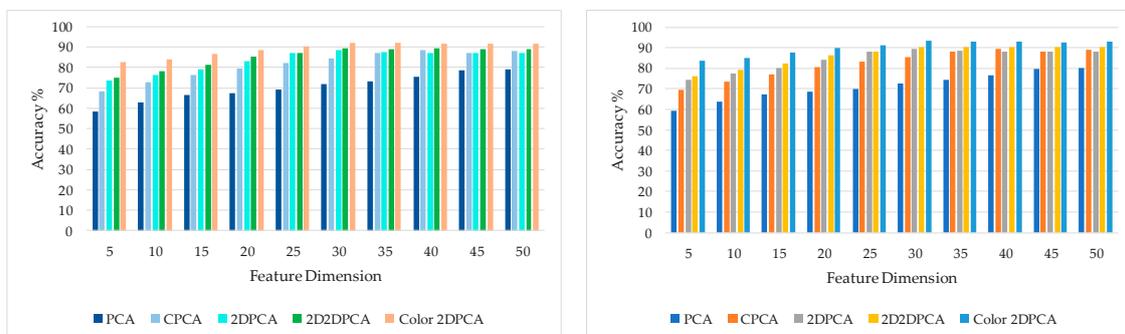


Figure 11. Evaluation results under various feature dimension in two benchmark databases: (a) FRGC; (b) LFW.

As can be seen from the experimental results shown in Figure 11, compared with other PCA dimensionality reduction algorithms, the recognition performance of the color 2DPCA algorithm is dominant in the two benchmark databases. Coincidentally, under different databases, the PCA and CPCA algorithms have the highest recognition accuracy when the dimension is 45, but the 2DPCA correlated algorithm tends to have the highest recognition accuracy when the dimension is 30, which can achieve the great recognition result at lower dimensions compared with the PCA-related algorithm. That is not surprising, color 2DPCA can effectively extract the color feature and spatial feature. Hence, in all the next experiments, the feature dimension will be set to 30.

3.3.3. Acquisition of the Optimal Similarity Weight Ratio in Different Benchmark Database

In this experiment, it was known that the similarity G was determined by Formula (28), and the optimal similarity weight Ω under different datasets was evaluated, respectively. FRGC face database and LFW database are selected as query image sets, and the color 2DPCA is trained and tested according to the proportion of 1.5:1. The size of face images is resized to 227×227 . There are nine groups of comparison experiments, respectively. The value of Ω values were: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. The experimental results are shown in Figure 12.

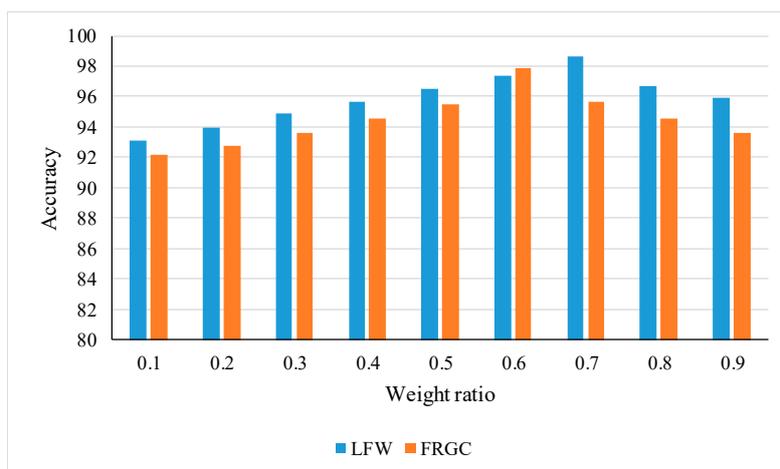


Figure 12. The face recognition rate of different similarity weights.

From the experimental results shown in Figure 12, for the LFW database, the optimal weight is $\Omega = 0.7$ and the recognition rate is 98.64%. For the FRGC face database, the optimal weight is $\Omega = 0.6$ and the optimal recognition rate is 97.86%. For different face database, the weight ratio is changing. When the weight ratio is maximum or minimum, i.e., decision-level fusion is at a disadvantage, the recognition rate is low. We can safely draw a conclusion that decision-level convergence is a great help to improve the recognition performance. Color 2DPCA can effectively extract color and spatial information from color face images, and still has good robustness for the test and training sets under the conditions of large differences, which can make up for the defect of poor recognition performance of the CNN network.

3.3.4. Acquisition of the Optimal Crop Size for Different CNN Models

In this experiment, the image size is related to both the recognition performance and computational cost of CNN network, to evaluate the performance of the CNN under different crop size, FRGC face database was selected as test dataset. There were five groups experiments: 127×127 , 160×160 , 192×192 , 227×227 , 250×250 . Due to the random data argumentation, the smaller the image size, the bigger the randomness in training. We compared the recognition performance of different image size under the FRGC protocol. The experimental results are shown in Table 4.

Table 4. The performance of the CNN with different crop sizes.

Crop Size	Accuracy %			
	CNN-0	CNN-0 + Fine-Tune	CNN-2	CNN-2 + Fine-Tune
127×127	64.32	67.96	83.24	84.01
160×160	73.94	76.31	87.14	87.97
192×192	85.75	87.19	93.77	94.37
227×227	89.12	90.23	97.86	98.02
250×250	86.63	88.27	95.34	95.15

As can be seen from the experimental results shown in Table 4, the optimal recognition performance was obtained when the pixel size is 227×227 .

3.3.5. Acquisition of the Optimal Distance Metric Method

In this experiment, we evaluated the influence of the distance metrics method for the proposed C2D-CNN method and expected to obtain the optimal combination of the distance metrics.

The benchmark database LFW, FRGC were selected as the test datasets. Four groups of experiments were carried out, the combinations were respectively: Mahal. Distance used only, Euclid. Distance used only, Euclid. Distance for CNN and Mahal. Distance for color 2DPCA, Mahal. Distance for CNN and Euclid. Distance for color 2DPCA. Since the distance metric algorithm has changed, the optimal similarity weight ratio has changed. According to the experiment, we obtained the optimal similarity weight ratio of the four distance algorithm combinations under two benchmark databases, 0.7/0.8 0.5/0.6, 0.5/0.6, 0.6/0.7, respectively. To embody the difference of the accuracy for various combinations more carefully, the experiment was performed under the condition of the best weight ratio. The results are shown in Figure 13.

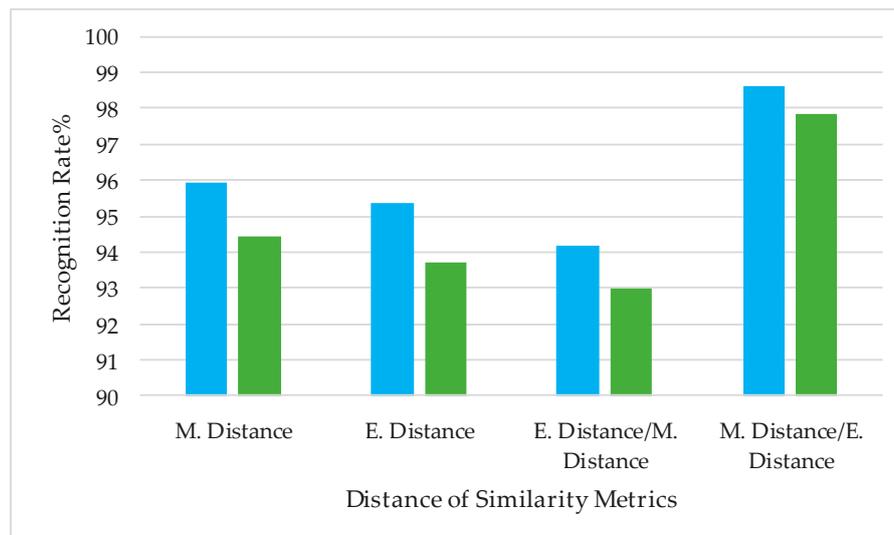


Figure 13. The accuracy under the different combinations of various distance metrics.

According to the experimental results shown in Figure 13, the combination of Euclidean distance for CNN features and Mahalanobis distance for color 2DPCA features can achieve optimal recognition performance. Different distance metric methods will lead to different recognition accuracy. Compared to Euclidean distance, Mahalanobis distance has a better performance for CNN features, and Euclidean distance is the most commonly used similarity metric method for 2DPCA and its related algorithms.

3.4. Testing Result and Discussion

In this part, the proposed C2D-CNN algorithm will be compared with other state-of-the-art algorithms to verify the effectiveness and robustness of our algorithm. Firstly, we will evaluate the validity of the proposed and introduced normalization layer, layered activation function, and probabilistic max-pooling, respectively. Secondly, we will return to the original intention of the proposed novel idea, the comparison of the fine-tuning and decision-level fusion. Finally, we will compare the proposed algorithm with the state-of-the-art method. The specific flow is shown in Figure 14.

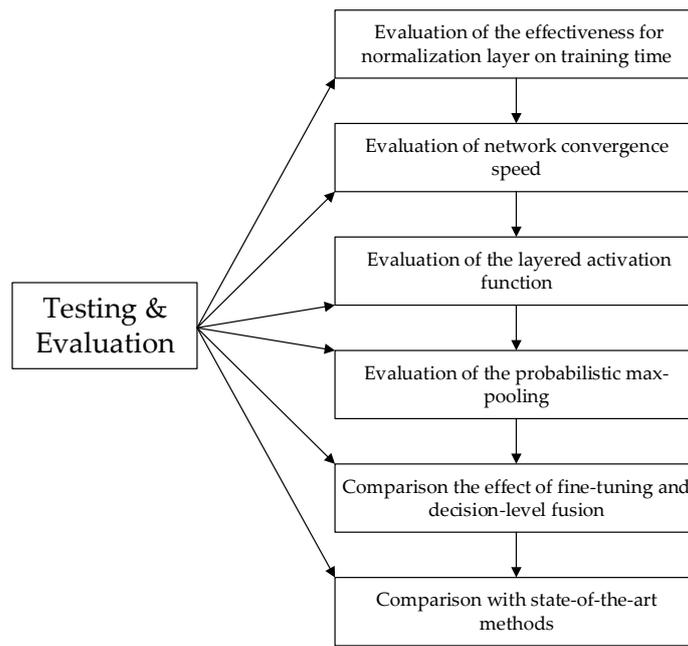


Figure 14. Flow chart of experiment.

3.4.1. Evaluation of the Performance of Normalization Layer on Training Time

In this experiment, we evaluated the effectiveness of normalization on training time. The overall network cost penalty was set as 0.01. There were three groups of experiments: CNN-0, CNN-1 and CNN-2. The learning rate of CNN-1 and CNN-2 was set as 0.05, and the learning rate of CNN-0 was set as 0.01. We set the value of weight ratio to be 0.6 on the FRGC database. The network was trained according to the experimental setup. The experimental results are shown in Figure 15.

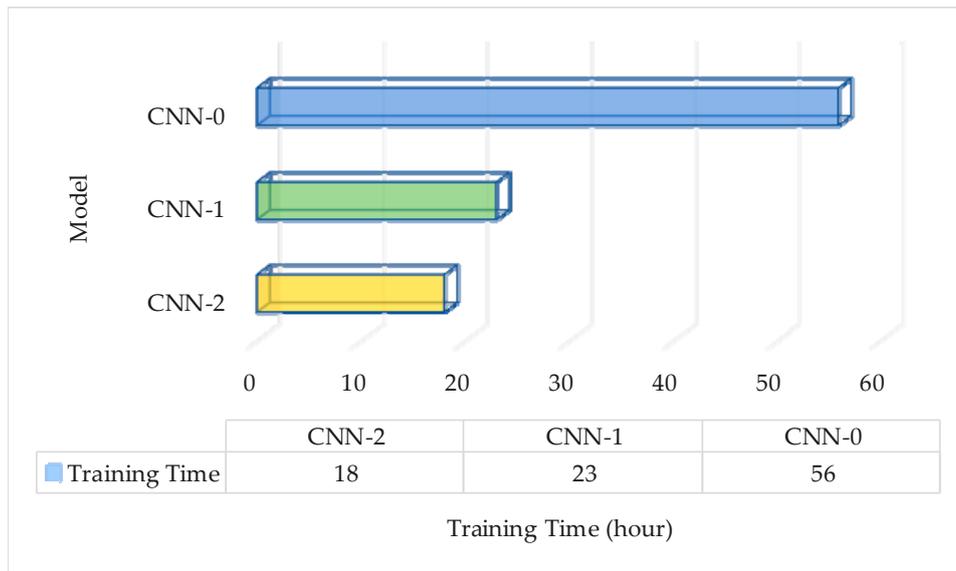


Figure 15. Comparison of the pre-training time of the three models.

As can be seen from the experimental results shown in Figure 15, the normalization layer demonstrated a good performance in the process of network training. The training time was reduced by 67.9% compared to the unnormalized CNN-0 network, and a 21.7% decrease compared to the

CNN-1 network for batch normalization. This is because the normalization layer can normalize data to standard normal distribution, which can effectively prevent overfitting, and can choose a larger learning rate to accelerate network convergence. Compared with batch normalization, the normalization has significant advantages. (1) The normalization layer removed the recovery operation, thus reducing the consumption of GPU memory and computation cost; (2) we used the mean and variance of the mini-batch data to estimate the mean and variance of all dimensions, linking them by updating the global variance and mean; (3) when the global variance and mean value were updated, the specific combination coefficient ϕ and ψ were learned from the data, and thus ϕ and ψ were self-adaptive to the input data.

3.4.2. Evaluation of Network Convergence Speed

In this experiment, three groups of experiments were carried out to evaluate the effects of the normalization layer on the convergence rate of the model. Compared with CNN-0, normalization and batch normalization were introduced to the CNN-1 and CNN-2 models, respectively. The CNN models with and without normalized process were pre-trained with the FRGC face databases, respectively. The network was trained according to the experimental setup. We set the value of weight ratio to be 0.6 on the FRGC database. According to the loss value, the convergence rate of the model can be evaluated. The experimental results are shown in Figure 16.

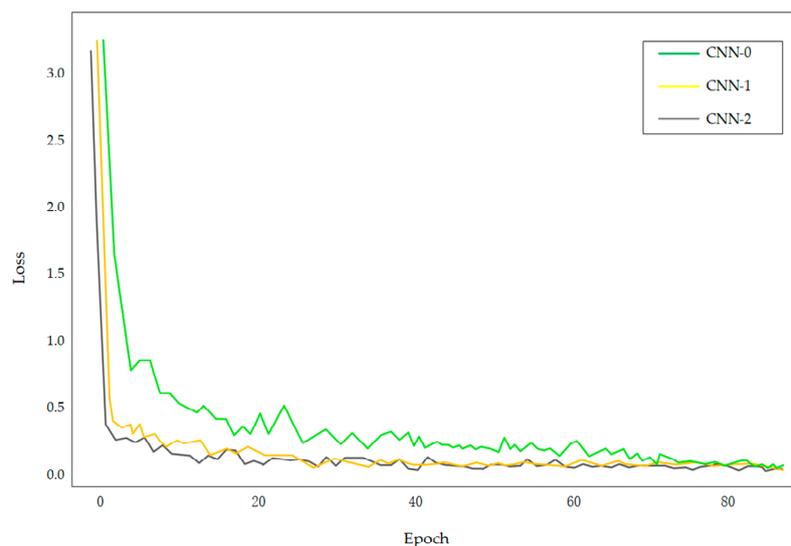


Figure 16. Comparison of the convergence speed of the two networks.

According to the experimental results shown in Figure 16, with the epoch, both of three models gradually converged, however, the convergence speed of the CNN-0 network without normalization is significantly slower than the CNN-1 and CNN-2 networks which were processed by normalization or batch normalization. Compared with CNN-1 network, although the two curves are close, the convergence speed of CNN-2 network still has obvious advantages. And as the iterative process proceeded, the loss value of the network tended to be stable. The experimental results verified that the normalization layer can effectively accelerate the convergence of the network.

3.4.3. Evaluation of the Performance of Layered Activation Function

To evaluate the activation performance of the layered activation function, we did experiments on two benchmark datasets. We selected the FRGC uncontrolled image and LFW database as the test set. The number of middle-level nodes in the layered activation function layer was set to three. We set the value of weight ratio to be 0.6 on FRGC database and 0.7 on the LFW database. Five groups of

experiments were performed: Sigmoid, ReLU, ELU, PRELU and LAF (layered activation function). We compared the effectiveness of five activation functions. Different activation functions will directly affect the recognition performance of the CNN model. In this experiment, we compared the recognition rate of the proposed C2D-CNN algorithm under different activation functions. The experimental results are shown in Figure 17.

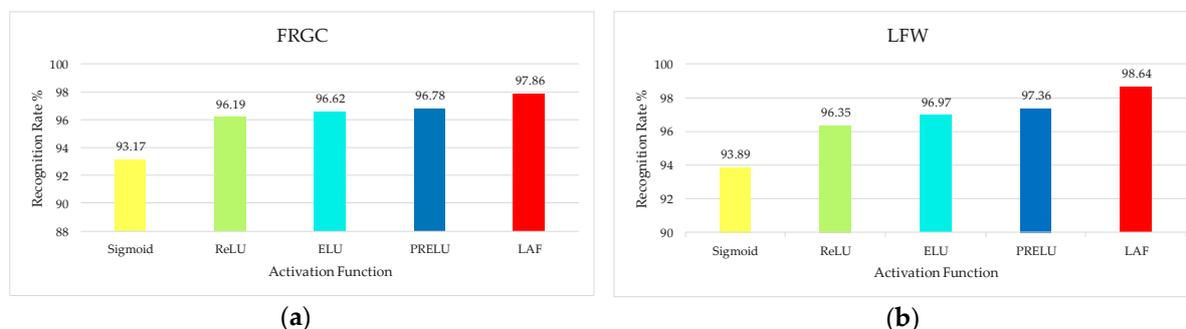


Figure 17. Comparison of activation functions: (a) FRGC face database; (b) LFW face database.

As is shown in Figure 17, identification performance of the network with layered activation function is better than the other basic activation functions. The success of the layered activation function is due to its adoption of a hierarchical acquisition strategy. The strategy of hierarchical activation, which introduces learning and adaptation into the activation operation to make it adapt to the normalized data, can give full play to the basic activation function and the role of the hierarchy. Firstly, by adaptively combining the basic activation function, we can solve the existing gradient problem well and get better activation effect. Then, the competition mechanism is introduced to enhance the non-linearity of the activation operation.

3.4.4. Evaluation of the Performance of the Probabilistic Max-Pooling

To evaluate the effectiveness of the pooling method on robustness and recognition performance, two benchmark databases, FRGC and LFW, were chosen for test set. In this experiment, we compared the accuracy of the proposed algorithm C2D-CNN under different pooling functions. We set the value of weight ratio to be 0.6 on FRGC database and 0.7 on the LFW database. The experiment includes three groups of comparison experiments: max-pooling, mean-pooling, and probabilistic max-pooling. The experimental results are shown in Figure 18.

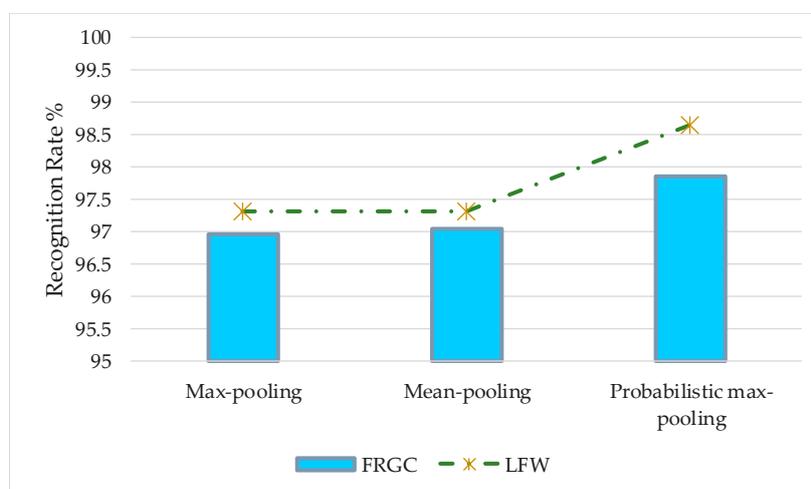


Figure 18. Comparison of pooling method.

As is shown on Figure 18, the performance of the network with probabilistic max-pooling is better than other pooling method and reached 97.86%. The mean and maximum of the neuron response values in the “accepted domain” are usually taken as the pooling rules. However, in the process of back propagation, the loss of feature representations is inevitable, which directly restricts the robustness of the CNN. A probabilistic max-pooling approach is applied so that the feature information is preserved to the maximum extent while maintaining feature invariance. Compared with the other pooling method, probabilistic max-pooling can achieve a better pooling performance.

3.4.5. Comparison of the Effectiveness of Fine-Tuning and Decision-Level Fusion

In this experiment, we evaluated the recognition performance of the C2D-CNN model and compared the effectiveness of decision-level fusion and fine-tuning in improving recognition accuracy. As there is no label information to use for fine-tuning the pre-trained model, we only test the performance on the FRGC database, and chose to experiment with FRGC Experiment 4. This included four groups of experiments: CNN-0, CNN-0 + fine-tuning, C2D-CNN, C2D-CNN + fine tuning; The similarity weight is set to 0.6. The experimental results are shown in Figure 19.

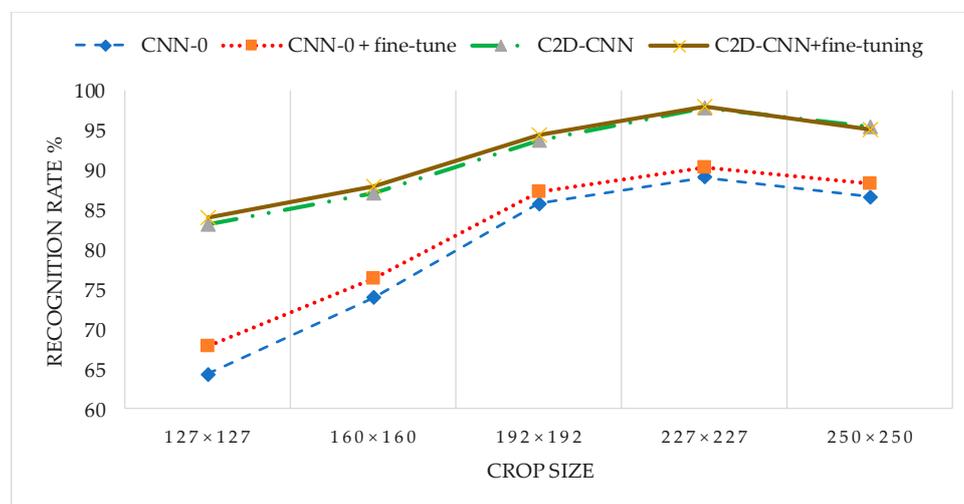


Figure 19. Evaluation the effect of fine-tuning and decision-level fusion on improving the performance of CNN.

As can be seen from the experimental results shown in Figure 19, the difference of recognition performance between the fine-tuned network and original network is very small, especially for the proposed model in this article, which is almost no difference. The experimental results show that when there is big difference between test and training set, fine tuning cannot solve the problem of low recognition rate well, but the introduction of the color 2DPCA feature is more effective than trimming CNN in improving the performance of pre-training the CNN model. This is because the C2D-CNN model based on decision-level fusion combines two features that have complementary properties which integrates the advantages of the color 2DPCA feature and CNN feature so as to make the fused similarity more accurate and generalization ability enhanced.

3.4.6. Comparison with State-of-the-Art Methods

We evaluated our method in the close-set and open-set face identification tasks, following the protocol in [58]. The close-set identification protocol reports the Rank-1 identification accuracy and the open-set identification reports the DIR (Detection and Identification Rate) at FAR (Detection and Identification Rate) equal to 1%. The comparisons with the state-of-the-art method are shown in Table 5. The proposed C2D-CNN model achieves state-of-the-art performance in face identification tasks.

Table 5. The performance of our method and state-of-the-art methods on LFW under the identification protocol.

Method	Rank-1 (%)	DIR@1% FAR (%)
COST-S1	56.7	25.0
COTS-S1 + S2	66.5	35.0
DeepFace	64.9	44.50
WSTFusion	82.5	61.90
Our Method	91.98	63.34

The performance results of the proposed method clearly demonstrated that C2D-CNN is very effective for the improvement of face recognition; what's more, the CNN model we proposed and applied has achieved optimal results in the process of identification and training. All of the architectures performed very well on two benchmark datasets. The diversity of these datasets demonstrates that our model is effective for a wide range of activity recognition tasks. The C2D-CNN method using decision-level fusion has a better recognition performance than the fine-tuning network when there are big differences between test set and training set in viewpoint, expression, illumination, or other factors.

There are two main reasons for the superb performance of the proposed method C2D-CNN. First, in the design of CNN network architecture, we fully consider the network training and identification process, and constantly optimize the architecture of CNN, proposed and introduced the normalization layer, layered activation function layer, and the probabilistic max-pooling, cross-entropy cost function, which has a good performance in improving the accuracy of face recognition and training speed. Second, before we came up with C2D-CNN, we considered that it was not enough to optimize the performance of recognition only from the improvement of network architecture. However, we could fundamentally solve the problems that CNN cannot avoid by fully considering the complementary information of different features. We consider two complementary features, which are fused at the decision level, and the experimental results show that this fusion is successful.

4. Conclusions and Future Work

In this paper, we proposed a novel robust face recognition using a deep C2D-CNN model under decision-level fusion to solve face recognition when there is a big difference between the test and the training set. Inspired by the idea of complementarity, two features are fused at the decision level. In addition, to improve the performance of CNN network, we proposed a novel network structure. Firstly, we introduced a normalization layer in CNN to speed up the network convergence and shorten the training time. Secondly, the layered activation function was introduced to make the activation function adaptive to the normalized data. Finally, probabilistic max-pooling was applied to preserve the feature representations in order to maximize extent while maintaining feature invariance. In the future, we will focus on the deeper integration of deep learning and traditional machine learning approaches to address face recognition problems in more severe environments.

Author Contributions: J.L. and T.Q. conceived idea and initialized research; they conceived and designed the experiments; C.W. performed the experiments; K.X. and F.-Q.W. analyzed the data and reviewed the paper; C.W. contributed reagents/materials/analysis tools; J.L. wrote the paper.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 61701046 and the Undergraduate Training Programs for Innovation and Entrepreneurship of Yangtze University under Grant 2017008.

Acknowledgments: Many people have made invaluable contributions, both directly and indirectly to our research. Particularly, I would like to express our gratitude to Guan-Qun Sheng and Ning-Jun Ruan for selfless help and advice. They provided many favorable suggestions for the grammar and experimental parts of my manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abate, A.F.; Nappi, M.; Riccio, D. 2D and 3D face recognition: A survey. *Pattern Recognit. Lett.* **2007**, *28*, 1885–1906. [[CrossRef](#)]
2. Kim, D.J.; Sohn, M.K.; Kim, H. Geometric Feature-Based Face Normalization for Facial Expression Recognition. In Proceedings of the 2nd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), Madrid, Spain, 18–20 November 2014; pp. 172–175.
3. Ouarda, W.; Trichili, H.; Alimi, A.M. Face recognition based on geometric features using Support Vector Machines. In Proceedings of the 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Tunis, Tunisia, 11–14 August 2014; pp. 89–95.
4. Wei, M.; Ma, B. Face Recognition Based on Randomized Subspace Feature. In Proceedings of the 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, 9–11 November 2015; pp. 668–674.
5. Chen, C.; Dantcheva, A.; Ross, A. An ensemble of patch-based subspaces for makeup-robust face recognition. *Inf. Fusion* **2016**, *32*, 80–92. [[CrossRef](#)]
6. Hanmandlu, M.; Gupta, D.; Vasikarla, S. Face recognition using Elastic bunch graph matching. In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 23–25 October 2013; pp. 1–7.
7. Chen, X.; Zhang, C.; Dong, F.; Zhou, Z. Parallelization of elastic bunch graph matching (EBGM) algorithm for fast face recognition. In Proceedings of the IEEE China Summit & International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 201–205.
8. Wan, L.; Liu, N.; Huo, H.; Fang, T. Face Recognition with Convolutional Neural Networks and subspace learning. In Proceedings of the 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017; pp. 228–233.
9. Qi, X.; Liu, C.; Schuckers, S. CNN based key frame extraction for face in video recognition. In Proceedings of the IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA), Singapore, 11–12 January 2018; pp. 1–8.
10. Liang, Y.; Zhang, Y.; Zeng, X.X. Pose-invariant 3D face recognition using half face. *Signal Process. Image Commun.* **2017**, *57*, 84–90. [[CrossRef](#)]
11. Li, Y.; Wang, Y.; Liu, J.; Hao, W. Expression-insensitive 3D face recognition by the fusion of multiple subject-specific curves. *Neurocomputing* **2018**, *275*, 1295–1307. [[CrossRef](#)]
12. Dagnes, N.; Vezzetti, E.; Marcolin, F.; Tornincasa, S. Occlusion detection and restoration techniques for 3D face recognition: A literature review. *Mach. Vis. Appl.* **2018**, 1–25. [[CrossRef](#)]
13. Vezzetti, E.; Marcolin, F.; Tornincasa, S. 3D geometry-based automatic landmark localization in presence of facial occlusions. *Multimed. Tools Appl.* **2017**, *14*, 1–29. [[CrossRef](#)]
14. Moos, S.; Marcolin, F.; Tornincasa, S.; Vezzetti, E.; Violante, M.G.; Fracastoro, G.; Padula, F. Cleft lip pathology diagnosis and foetal landmark extraction via 3D geometrical analysis. *Int. J. Interact. Des. Manuf.* **2017**, *11*, 1–18. [[CrossRef](#)]
15. Moeini, A.; Faez, K.; Moeini, H. Face recognition across makeup and plastic surgery from real-world images. *J. Electron. Imaging* **2015**, *24*, 053028. [[CrossRef](#)]
16. Ramalingam, S. Fuzzy interval-valued multi criteria based decision making for ranking features in multi-modal 3D face recognition. *Fuzzy Sets Syst.* **2018**, *337*, 25–51. [[CrossRef](#)]
17. Abbad, A.; Abbad, K.; Tairi, H. 3D face recognition: Multi-scale strategy based on geometric and local descriptors. *Comput. Electr. Eng.* **2017**. [[CrossRef](#)]
18. Kim, J.; Han, D.; Hwang, W.; Kim, J. 3D face recognition via discriminative keypoint selection. In Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 28 June–1 July 2017; pp. 477–480.
19. Soltanpour, S.; Wu, Q.J.; Anvaripour, M. Multimodal 2D-3D face recognition using structural context and pyramidal shape index. In Proceedings of the 6th International Conference on Imaging for Crime Prevention and Detection (ICDP), London, UK, 15–17 July 2015; pp. 1–6.
20. Wang, X.; Ly, V.; Guo, R.; Kambhamettu, C. 2D-3D face recognition via restricted boltzmann machines. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, 5–8 January 2014; pp. 574–580.

21. Wang, X.; Ly, V.; Guo, G.; Kambhamettu, C. A new approach for 2d-3d heterogeneous face recognition. In Proceedings of the IEEE International Symposium on Multimedia (ISM), Anaheim, CA, USA, 9–11 December 2013; pp. 301–304.
22. Kakadiaris, I.A.; Toderici, G.; Evangelopoulos, G. 3D-2D face recognition with pose and illumination normalization. *Comput. Vision Image Underst.* **2017**, *154*, 137–151. [[CrossRef](#)]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
24. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
25. Dong, C.; Loy, C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Swiss, 6–12 September 2014; pp. 184–199.
26. Cai, H.; Yan, F.; Mikolajczyk, K. Learning weights for codebook in image classification and retrieval. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2320–2327.
27. Chaib, S.; Yao, H.; Gu, Y.; Amrani, M. Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models. In Proceedings of the Ninth International Conference on Digital Image Processing (ICDIP), Hongkong, China, 19–22 May 2017; p. 104203D.
28. Liu, Y.; Li, Y.; Ma, X.; Song, R. Facial Expression Recognition with Fusion Features Extracted from Salient Facial Areas. *Sensors* **2017**, *17*, 712. [[CrossRef](#)] [[PubMed](#)]
29. Lu, J.; Wang, G.; Zhou, J. Simultaneous feature and dictionary learning for image set based face recognition. *IEEE Trans. Image Process.* **2017**, *26*, 4042–4054. [[CrossRef](#)] [[PubMed](#)]
30. Hu, G.; Peng, X.; Yang, Y.; Hospedales, T.M.; Verbeek, J. Frankenstein: Learning deep face representations using small data. *IEEE Trans. Image Process.* **2018**, *27*, 293–303. [[CrossRef](#)] [[PubMed](#)]
31. Oquab, M.; Bottou, L.; Laptev, I. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1717–1724.
32. Masi, I.; Trần, A.T.; Hassner, T.; Leksut, J.T.; Medioni, G. Do we really need to collect millions of faces for effective face recognition? In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 579–596.
33. Campos, V.; Jou, B.; Giro-i-Nieto, X. From pixels to sentiment: Fine-tuning cnns for visual sentiment prediction. *Image Vis. Comput.* **2017**, *65*, 15–22. [[CrossRef](#)]
34. Nagori, V. Fine tuning the parameters of back propagation algorithm for optimum learning performance. In Proceedings of the 2nd International Conference on contemporary Computing and Informatics, Noida, India, 14–17 December 2016; pp. 7–12.
35. Tzelepi, M.; Tefas, A. Exploiting supervised learning for finetuning deep CNNs in content based image retrieval. In Proceedings of the 23rd International Conference on the Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2918–2923.
36. Li, Y.; Xie, W.; Li, H. Hyperspectral image reconstruction by deep convolutional neural network for classification. *Pattern Recognit.* **2016**, *63*, 371–383. [[CrossRef](#)]
37. Rawat, W.; Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput.* **2017**, *29*, 2352–2449. [[CrossRef](#)] [[PubMed](#)]
38. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.
39. Shang, W.; Sohn, K.; Almeida, D.; Lee, H. Understanding and improving convolutional neural networks via concatenated rectified linear units. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 2217–2225.
40. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
41. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International conference on machine learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
42. Choi, J.Y.; Ro, Y.M.; Plataniotis, K.N. Color local texture features for color face recognition. *IEEE Trans. Image Process.* **2012**, *21*, 1366–1380. [[CrossRef](#)] [[PubMed](#)]

43. Lu, Z.; Jiang, X.; Kot, A. A novel LBP-based Color descriptor for face recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1857–1861.
44. Lu, Z.; Jiang, X.; Kot, A. An effective color space for face recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 849–856.
45. Lu, Z.; Jiang, X.; Kot, A. A Color Channel Fusion Approach for Face Recognition. *IEEE Signal Process. Lett.* **2015**, *22*, 1839–1843. [[CrossRef](#)]
46. Qian, S.; Liu, H.; Liu, C. Adaptive activation functions in convolutional neural networks. *Neurocomputing* **2017**, *272*, 204–212. [[CrossRef](#)]
47. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 609–616.
48. Beigy, H.; Meybodi, M.R. Adaptation of Parameters of BP Algorithm Using Learning Automata. In Proceedings of the Sixth Brazilian Symposium on Neural Networks, Rio de Janeiro, Brazil, 22–25 November 2000; pp. 24–31.
49. Kline, D.M.; Berardi, V.L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Comput. Appl.* **2005**, *14*, 310–318. [[CrossRef](#)]
50. Xuelong, L.; Yanwei, P.; Yuan, Y. L1-norm-based 2DPCA. *IEEE Trans. Syst. Man Cybern.* **2010**, *40*, 1170–1175. [[CrossRef](#)] [[PubMed](#)]
51. Moore, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Autom. Cont.* **1981**, *26*, 17–32. [[CrossRef](#)]
52. Xinguang, X.; Yang, J.; Qiuping, C. Color face recognition by PCA-like approach. *Neurocomputing* **2015**, *152*, 231–235. [[CrossRef](#)]
53. Saxena, S.; Verbeek, J. Heterogeneous face recognition with CNNs. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 483–491.
54. Lu, Z.; Jiang, X.; Kot, A. Feature fusion with covariance matrix regularization in face recognition. *Signal Process.* **2018**, *144*, 296–305. [[CrossRef](#)]
55. Perlibakas, V. Distance measures for PCA-based face recognition. *Pattern Recognit. Lett.* **2004**, *25*, 711–724. [[CrossRef](#)]
56. Xu, A.; Jin, X.; Jiang, Y.; Guo, P. Complete two-dimensional PCA for face recognition. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR), Hong Kong, China, 20–24 August 2006; pp. 481–484.
57. Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; No. 2. Technical Report 07-49; University of Massachusetts: Amherst, MA, USA, 2007; Volume 1.
58. Phillips, P.J.; Flynn, P.J.; Scruggs, T. Overview of the Face Recognition Grand Challenge. In Proceedings of the Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 947–954.

