Journal of Brilliant Engineering 3 (2022) 4563
Brilliant Engineering
journal home: www.acapublishing.com/journals/4/ben

Research Article

Improved Reliable Deep Face Recognition Method Using Separated Components

Hossein Chamani[1], Mohammad Nadjafi[*,2]
[1]Master of Computer Engineering, Tehran, Iran
[2]Assistance Professor of Aerospace Engineering, Aerospace Research Institute (Ministry of Science, Research and Technology), Tehran, Iran

Keywords

Deep learning,
Convolutional neural network,
Machine vision,
Facial components segmentation,
Facial recognition.

Abstract

Face recognition is used as one of the most successful biometric methods due to the availability of advanced resources such as faster processors and higher memory and providing intelligent methods based on the power of these resources. Nevertheless, there are still many challenges in this area. The face plays an important role in the transmission of emotions and carries the characteristics hidden in it, the identity of individuals. Face recognition has been added to some control devices, security, welfare, criminal identification, and many other areas, which is the main motivation for research in this field. In this paper, the DCSFR method is presented to pay attention to the main features of the face such as eyes, lips, mouth, and nose, which is the main novelty of this work, to get higher accuracy or speed than the previously existing methods. In this approach, instead of using general information in face recognition, facial components such as eyes, nose, mouth are separated into another image, and face classification operations (deep learning by convolution neural network) are performed on separated components. The results show that the computational cost with the proposed method is reduced by about 70%. Also, it can be achieved that CNN does not perform as well as the complete picture of the disassembled components.

1. Introduction

Research into automatic face recognition has been around since the 1960s. Today, the need for techniques and methods of identifying the identity and feelings of people by computer is increasing. Since recognizing faces is a natural process because humans usually recognize each other effortlessly and without error. On the other hand, using face recognition in machine vision, pattern recognition, and image processing remains a challenging issue. The parameters involved in face recognition have made it possible to identify faces using a machine automatically.

Face recognition is one of the fastest ways to identify a person. There is not much delay in obtaining a face image. In many cases, people are unaware of the process and do not feel insecure. Important applications of face recognition that increase the need for research in this area include payments, access and security, criminal identification, and advertising [1]. Researchers have recently found that the broad generalization of face recognition systems increases their vulnerability to attacks. Thus, image-based attacks that use the morphing technique for input pose a serious security risk to face recognition systems. In this regard, research has been presented to deal with morphing techniques using facial landmarks [2]. Of course, this technique should not be confused with face-averaging. The difference between these two techniques is given in [3].

2. Research background

In order to identify a face, the features of each face must be extracted. Several methods for feature extraction, such as Gabor filters, local binary patterns, feature bags, and Fisher vectors, can model decision boundaries nonlinearly. The various methods of extracting facial features cause separate face recognition algorithms. Face image processing for face recognition can be divided into specific categories: comprehensive, structural-based methods; combinatorial/hybrid methods; and methods based on neural networks and deep learning (as shown in Figure 1).

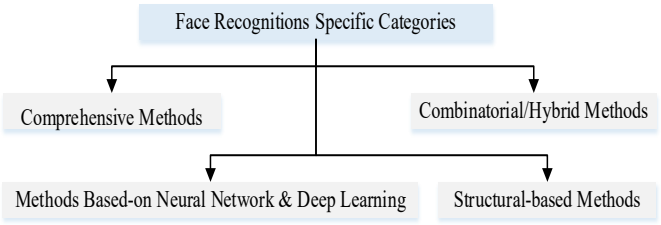

Figure 1. Specific methods of face recognition

Many factors effectively identify the face [4], the most important: facial expressions, the image light, head position, current image age and aging, non-fundamental changes such as whether or not to use glasses, beard, makeup, etc. Important features of a face can be expressed eyes, nose, lips, eyebrows, length and width of cheeks and forehead, skin color, beard, and mustache. Figure 2 shows the challenges of face recognition.

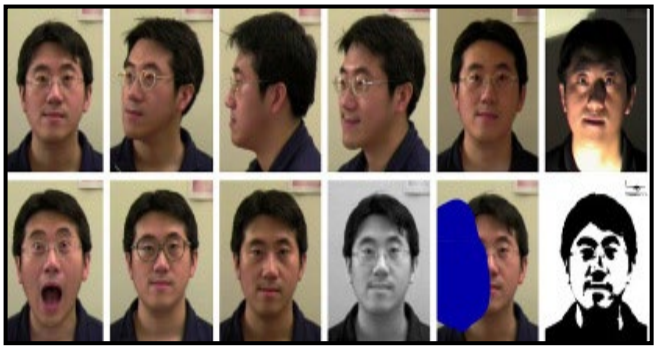

Figure 2. Face recognition challenges

*Corresponding Author: m.nadjafi@ari.ac.ir
Received 17 October 2021; Revised 15 Jan 2022; Accepted 15 Jan 2022
https://doi.org/10.36937/ben.2022.4563

Face images as an array of large pixels often belong to several categories of small pixel arrays. Algebraic and statistical tools for extracting and analyzing these arrays have been provided so far. In fact, comprehensive methods (based on appearance) use the whole image and pixel values to compare faces and extract features and estimate the comprehensive pattern. Comprehensive methods can be categorized into 1. linear 2. nonlinear.

The Eigenface method was introduced in 1990 as a method for simple face-feature calculation. For the first time, this algorithm was used to identify faces [5]. Then, different versions such as Modular PCA [6], 2DPCA [7] were presented to eliminate the shortcomings of this method. The LRC method [8] and the LDA method [9] are other linear methods. Due to the limitations of linear methods, nonlinear methods that work comprehensively on face images were proposed. KPCA [10], KLDA [11], and LLE [12] are nonlinear algorithms in which the main idea is to transmit or map high-dimensional face-to-space images, in which faces can be expressed linearly. Moreover, thus, used linear methods to separate classes.

Structure-based methods implemented based on feature search have special capabilities over comprehensive linear or nonlinear methods. These are methods that try to find the immutable features of a face despite the head's angle or change of direction. The immutable features of this method are the features that always exist, even when there are some internal or external conditions. There is another well-known algorithm that solves the problem of face recognition using the face structure, the LBP algorithm [13]. The LBP algorithm expresses the change of neighborhoods concerning a central pixel. Various other methods have been proposed to improve and strengthen the structure-based methods modeled on the LBP algorithm [14-16]. Another structure-based method is the 3D model. The input of 3D algorithms can be images obtained from 3D scanners or normal RGBD images. In some cases, to identify faces under different lighting, 3D models are combined with the same spherical lighting [17] (Figure 3).

Figure 3. An example of a 3D pre-processed face image for face recognition

The third category is called hybrid methods, algorithms that use a combination of several linear, nonlinear, structure-based methods, neural networks, and so on [18]. The SOM network is not good when slight changes in the image sample, while the convolution networks show is flexible. In [19], the authors presented a possible decision-based neural network model for three different applications (face positioning, eye positioning, and face recognition) and provided relatively better performance. A hybrid algorithm was also introduced in [20], in which, through PCA, features are extracted and used as input to an RBF neural network. RBFs have a strong topology and high approximation power, and learning in RBFs is fast.

With the rapid development of deep learning in recent years, face recognition has seen remarkable success. In fact, the main power of deep neural networks is in the functions used in different intermediate layers inside neurons. Deep-learning methods attempt to use input data hierarchically, and most of them, like CNN, is modeled on the neural network structure of animals and humans. The main feature of these methods is using a convolution network feature extractor. Deep Face is one of the most successful works in using a deep neural network to identify faces [21-26].

In face processing, the separation of facial components is also called "face landmark detection." Recognizing facial landmarks has many applications, such as face recognition, head direction estimation, face transformation, virtual makeup, and face movement. Recognition of facial landmarks is still limited in the real world, even under ideal conditions. This limitation is due to large background changes, facial expressions, facial obstructions, and brightness [24]. The input of a landmark detection algorithm is usually an image represented by a face detection algorithm.

3. Proposed DCSFR method

Extracting facial image features is an important step in identifying and classifying faces. In all previous works – with our knowledge - which has used deep learning to identify the face, the input image has been used, regardless of the importance of the various parts of the face such as eyes, nose, nose, mouth, etc. It is true that these methods somehow extract the physical features of the face, but none of the mentioned methods have given more importance to the important parts of the face. Different people can have many similarities in different parts of the face compared to important parts such as eyes, nose, mouth, etc. For this reason, we intend to propose the DCSFR method, in which instead of using the raw image as input, we separate the face components and use the isolated face components as the main input.

3.1. Separation of face components

To separate the components of the face, we use the landmarks of the face in this work. The face landmarks are actually (x, y) that indicate the position of different points of the face. Face landmarks are used for face positioning and many tasks in face processing. The number of landmarks varies depending on the method used to extract them. The method presented in [27] detects landmarks in three cases of 29, 194, and 68. Figure 3 shows an example of facial landmarks. The components of the face that we intend to separate in this work are the left eye, right eye, nose, and mouth, with about a 10% margin. The margin is to ensure that the components retain important information.

Figure 4. An example of facial landmarks

Each landmark has its label for better understanding and better recall. If fewer components are used, the number of filters that we will use in the next step after separating the face components - that is, using the convolution network - will be reduced. This is because filters of different sizes must be used to identify the features of different areas, such as around the face and chin. In the following, important landmarks in the separation of each component of the face are introduced. Figure 5 shows the process of our proposed method.

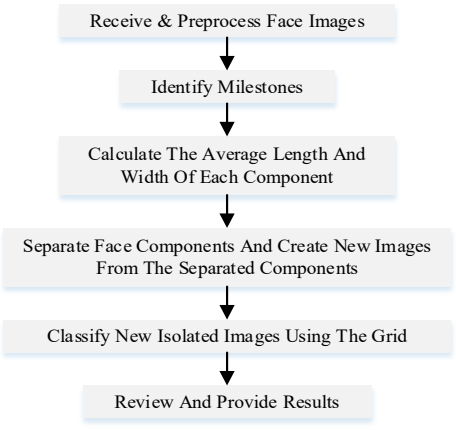

Figure 5. Flowchart of the proposed method

Features of different parts such as around the face and chin, filters of different sizes should be used. In the following, important landmarks in the separation of each component of the face are introduced.

Left eye landmarks: To separate the left eye, we will use four landmarks to indicate the length, width, and height of the left eye. These points are as follows: left_eye_bottom: The left eye's lowest (x, y). left_eye_left_corner: The leftmost (x, y) of the left eye. left_eye_right_corner: The most right (x, y) of the left eye. left_eye_top: The left eye's highest (x, y) (Figure 6).

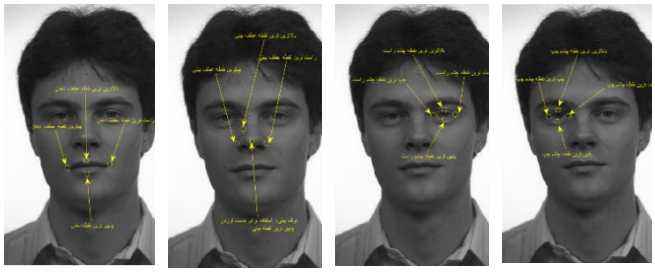

Figure 6. Important landmarks for extracting face components

Different parts of the face in the image can be at different angles, in this case, the distance between the leftmost point (left_eye_left_corner) and the right point of the left eye (left_eye_right_corner) by drawing a vertical line from the left point on the screen and measuring the distance from the right point to it, length of the left eye is obtained. Also, to obtain the width of the left eye, the distance between the highest point of the left eye (left_eye_top) and the lowest point of the left eye (left_eye_bottom) was used. Using the distance plane means using the coordinate distance of two points on only one axis.

Landmarks of the right eye: Measuring the length and width of the right eye is the same as the left eye, except that the leftmost point of the right eye is the inside. The four landmarks used to indicate the right eye's length, width, and height are as follows: right_eye_bottom: The lowest (x, y) of the right eye. right_eye_left_corner: The leftmost (x, y) of the right eye. right_eye_right_corner: The most right (x, y) of the right eye. right_eye_top: The right eye's highest (x, y) (Figure 6).

Nose landmarks: The third organ that must be separated is the nose. The points used to extract the nose are as follows: nose_contour_left1: As the highest point of the nose. nose_right: as the straightest point of the nose. nose_left: as the leftmost point of the nose. nose_tip: as the lowest landmark of the nose (Figure 6).

Mouth landmarks: The last element to be separated is the mouth. The mouth also includes the lower lip and upper lip and can be the largest element of the face. The landmarks used to separate the mouth are as follows: mouth_left_corner: The mouth's leftmost (x, y). mouth_right_corner: The most right (x, y) of the mouth. mouth_upper_lip_top: The highest (x, y) related to the mouth. mouth_lower_lip_bottom: The mouth's lowest (x, y) (Figure 6).

All detached components will be different from their counterparts in the same class. For example, a person's eyes can be open, semi-open, and closed. The mouth can be in the state of chewing, laughing, whistling, etc. In addition to this intra-class difference, inter-class differences exist between components. For example, the size of the components can vary, with one person having a long nose and the other wide. To better understand, the eyes and mouth of a Chinese person can be considered with the eyes and mouth of an Iranian person, how these components are inherently different between two different people.

Since they are used in the convolutional network in the next step after separating the face components, the size of the input images must be the same. These make differences between the classes, and the classes make the input of the convolutional network different in size. There is a noticeable difference in the size of the images. A W window is used for each component to separate to solve this problem. Thus, each face component's average length and width are obtained by obtaining the average landmarks. The following is the formula for calculating the average landmarks to separate the components of the face.

Calculating the mean of the leftmost x for the left eye: Since each of the detected landmarks of the face is represented by (x, y), we will use the x of each landmark to determine the left and right points of the various components, and therefore to calculate the left eye, we have a left eye point:

$$LELC = \frac{1}{m}\left(\sum_{m}^{i=1} LELC_{(i,1)}\right) \quad (1)$$

Where (i, 1) refers to the first element of the element m i. The first element is the x and the second element is the y landmark. Here is the leftmost point of the left eye. LELC is the same as left_eye_left_corner. Calculate the mean of the rightest x for the left eye:

$$LERC = \frac{1}{m}\left(\sum_{m}^{i=1} LERC_{(i,1)}\right) \quad (2)$$

Where LERC is the same as left_eye_right_corner. Calculate the mean of the highest y for the left eye:

$$LET = \frac{1}{m}\left(\sum_{m}^{i=1} LET_{(i,2)}\right) \quad (3)$$

Where LET is the same as left_eye_top. Calculate the mean of the lowest y for the left eye:

$$LEB = \frac{1}{m}\left(\sum_{m}^{i=1} LEB_{(i,2)}\right) \quad (4)$$

Where LEB is the same as left_eye_bottom.

After finding the average of all points, the length and width of the left eye can be calculated, and the left eye can be separated for all images. The output will be a left-eye image with the same length and width for all face images. Displays examples of isolated components of each face. The same is done to separate the other components.

Figure 7. An example of the isolated components of each face

The output of the separated components is stored in a new matrix called MatOfComponents, which is saved as a file for later use. The new images will be used as a replacement for the original image for the convolution network input.

3.2. Using deep convolution network

CNN uses filters on the pixels of any image to learn detailed patterns compared to global patterns with a traditional neural network. In fact, ConvNet is a class of neural networks specializing in processing data with a grid-like topology, such as an image [28]. The convolution network consists of different input layers, filters, convolution layers, rectifier layers, aggregation layers, and the fully connected layer in the last layer. The last layer can have more or fewer neurons in the number of available classes, and each of these neurons points individually to each known and existing class. Neuron connections are weighted relative to the input with a loss function in the training phase. In this work, the weighting neurons is not the main idea but the extra layers, custom network layer layout; as mentioned in the next section, we modified the network layers to match DCSFR requirements. In the implementation section, an attempt has been made to use this network to identify faces using images of extracted components as input. We also did this experiment by separating components in CNN itself by adding extra weights to the important landmarks and their neighbors by filtering pane size in person

separately using an extra layer (extra layer). First, the datasets used for testing are introduced.

4. Implementation of DCSFR method

This section details the proposed method. We spoke with dermatologists and cosmetologists and asked them to help us select the most important face components that can differentiate from person to person. We have separated these components and used those to feed the neural network, as we mentioned earlier.

4.1. Introducing data

The introduced datasets are well-known and widely used databases in machine learning for face recognition, and each of them is a reference database in this field. These datasets are the Feret database, FEI database, ORL, and Georgia. The FERET database contains 7810 images of 725 people. One of the advantages of this database is the Aging feature (i.e., the existence of instances of a class at different time intervals). The weakness of this data is the difference between the number of instances of each class so that one class has only four instances while the other class has more than 50 instances. To provide almost the same amount of learning and test data for convolutional network input, the database is divided into several categories, and attempts are made to test classes whose sample size is close to each other.

Figure 8. An example of the FERET database image

The Feret database has two color and gray versions, in which the Color Feret version is used for experiments (Figure 8). Of course, before using this data set, we turned it into a gray image (due to the completeness of the samples in the color version).

The ORL database consists of 400 images of 40 people, with exactly ten images per person. All photos of a sample are taken at one time. Images were taken in "face positions" and "head orientation" are different (Figure 9).

Figure 9. An example of a data face image

The Georgia database consists of 750 images from 50 different people, 15 images per person. This database also has an Aging feature. For different images, different "face positions," "head orientation," and "changes in lighting conditions" are considered. (Figure 10).

Figure 10. An example of a Georgia face image database

The FEI database is a Brazilian face database containing 2800 images, 14 images for every 200 people. All images are in color and against a white background that rotates up to 180 degrees. All samples turn gray before testing (Figure 11).

Figure 11. An example of an FEI dataset face image

4.2. Implementation Details

Various configurations have been considered to implement the proposed DCSFR method, from the variety in the type of extraction of face components to the number and type of different intermediate layers in the deep convolution network. MATLAB software has been used for testing to implement the proposed method and test it on the introduced datasets. The system specifications are Intel Core-i3, 4GB RAM at 1600mhz, and EVO 860 SSD (read speed when using Data Store).

4.3. Classification Using Convolution Network

New images are used as new input instead of the original images to evaluate accuracy and cost. Of course, to compare, the face is also identified using the original images, new images that contain components of the face instead of the face itself are smaller in size. For this reason, we have tried to test different types of filters in convolution layers, especially the size of small filters. Due to the lack of extra space on the four sides of the new image, there is no need to use a large plane in the convolution layer for the filters. The next section presents the test results for the DCSFR method.

4.4. Test Results

This section presents the results obtained from the experiments performed for the four datasets introduced with different configurations. The size of the input images for the deep convolution network will change after separating the face components, each of which is given in the relevant subsection. The results obtained from the experiments performed for the Ferret dataset can be seen in Table 1. It should also be noted that the size of the new images used for the deep convolution network input is 33 x 35 pixels. New images include only parts of the face instead of the whole image. Experiments show that face recognition accuracy is increased when the number of class samples is close to each other.

The architecture of the proposed CNN is as follows:

Layer 1: image input [VARIABLE INPUT SIZE].

Layer 2: 2d convolution [VARIABLE FILTER SIZE, VARIABLE FILTER COUNT, 'PADDING,' VARIABLE VALUE FOR PADDING].

Layer 3: batch normalization.

Layer 4: rectified linear activation function.

Layer 5: downsampling using max pooling.

Layer 6: 2d convolution [doubled VARIABLE FILTER SIZE, VARIABLE FILTER COUNT, 'PADDING,' VARIABLE VALUE FOR PADDING].

Layer 7: batch normalization.

Layer 8: rectified linear activation function.

Layer 9: downsampling using max-pooling (Extra Layer for adding extra weight (experiment results with this layer are not included in the main result table)).

Layer 10: 2d convolution [doubled VARIABLE FILTER SIZE, VARIABLE FILTER COUNT, 'PADDING,' VARIABLE VALUE FOR PADDING].

Layer 11: batch normalization.

Layer 12: rectified linear activation function.

Layer 13: fully connected.

Layer 14: Softmax function.

Layer 15: final classification.

Table 1. Experiments performed for the FERET database

Number of samples	Number of selected classes for learning and testing	Test Accuracy	Train/ Learning Accuracy	% of Test Data
12-15	40	80.3%	97%	10%
4-6	50	58%	100%	1 sample
21-24	13	92%	98%	10%
16-19	23	83.4%	99%	10%
4-50	725	60%	98%	1 sample

ORL Dataset Due to having regular samples and classes regarding the number and type of images taken, not many changes were made to the test data. Many experiments have been performed on this dataset, and the results obtained from implementing the proposed method on the ORL dataset can be seen collectively in Table 2.

The Georgia Tech database, as mentioned, has regular images in terms of the number of samples and the type of image taken. By separating the face components, the size of the new images used as the input of the deep convolutional network became 52 x 54 pixels. The results obtained for this dataset using the proposed method can be seen in Table 3.

The FEI dataset has been used in various experiments in different experiments because of its high rotation samples. These samples are placed in each class with a specific number. Table 4 shows the results of the experiments performed. Some of the images in this data have a 90-degree angle of rotation relative to the camera, and the so-called existing image is a profile image. In some experiments, it has not been used to examine the performance of the introduced method in more detail.

Table 2. Results of experiments performed on the ORL database

Number of samples	Number of selected classes for learning and testing	Test Accuracy	Train/ Learning Accuracy	% of Test Data
10	40	94%	97%	10%
10	40	92%	98%	20%
10	40	91%	95%	25%

Table 3. Results of experiments performed on Georgia Tech database

Number of samples	Number of selected classes for learning and testing	Test Accuracy	Train/ Learning Accuracy	% of Test Data
15	50	94.5%	98.5%	8%
15	50	92%	98%	16%
15	50	89%	97.5%	24%
15	50	85%	95%	32%

Table 4. Results of experiments performed on FEI database

Description	Sample	Learn Test class	Test Accur-acy	Train Accur-acy	% of Test Data
Images that have more angle to the camera and/or the face image is not fully captured were not used in this test. The selection of test data was made randomly.	9	200	92.5%	98%	15%
	9	200	89.5%	99%	30%

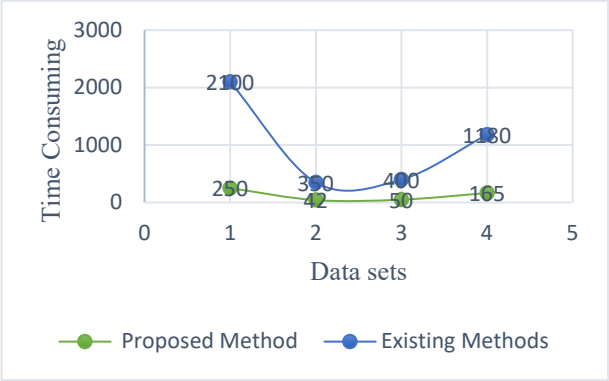

Figure 12. Average time spent learning and testing the Feret, Georgia Tech, The ORL, and FEI datasets for DCSFR and existing methods (L2-constrained-Softmax-Loss, Deep ID)

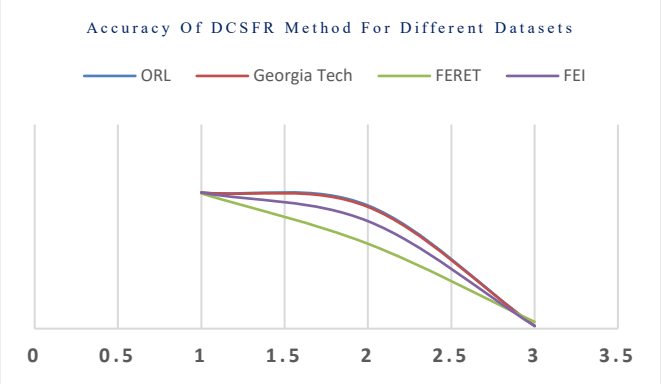

Figure 13. Mean learning and test accuracy obtained using the DCSFR method for different datasets

Types of experiments mean using different samples to learn and test. It should also be noted that each type of experiment is the average of other experiments with different configurations for the CNN network. The existing methods that have been included in the comparison (Figure 12) consist of L2-constrained-Softmax-Loss, and DeepID methods. The average time that these methods took for training and testing datasets used for this work is in this figure. These two methods, which are state-of-the-art methods for face recognition, have more time cost for training and testing than DCSFR.

5. Conclusion and future work

Face recognition is currently being researched in various fields of image processing, machine vision, pattern recognition, and many other areas, and many methods have been proposed for it. The proposed methods for face recognition have moved more towards deep learning due to the power of deep learning methods. One of the most successful deep learning methods for face recognition is using a deep convolutional neural network. Since different versions and configurations of this network have been used to identify faces, a new

method called DCSFR was introduced in this paper. Before using the convolution network to classify and identify faces, the main components of faces are identified using face landmarks. Separated and then put together to create a new image. The newly created image was then used to input the deep convolutional network. The proposed method was tested with popular and widely used face recognition datasets. An important result of the experiments is that due to the size of the input image, the time spent learning and testing the DCSFR method is reduced by about 70% compared to the method that does not use the separation of facial components. Another important observation is that despite the use of different sizes of filters, especially the small size of the filter and despite the use of a different number of filters, as well as different layers of aggregation and a different number of layers of CNN, the segmented face image used as input does not work as good as a full image and CNN works properly to identify the face when the full-face image is given as input to the convolution network (1-5 percent variation that depends on the dataset). Of course, we will check out other parameters to ensure this. In the future, we intend to pay attention to the head's direction of rotation. The head's direction of rotation is important because the DCSFR method uses an average length and width of the components to separate the face components. This may cause some component information whose length and width differ from the mean to be lost. We also intend to examine methods other than point averaging in more detail so that we may be able to prevent the loss of isolated component information. We intend to use our method, DCSFR, in addition to the current datasets, on the MEGA Face database, which has about 5 million images of about 672,000 identities, and examine the result because we believe that the deep convolution network can achieve better results with providing a large number of samples, even with the DCSFR method.

Declaration of Conflict of Interests

The authors declare that there is no conflict of interest. They have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1.] Jain, A.K. and S.Z. Li, Handbook of face recognition. 2011: Springer.

[2.] Scherhag, U., et al. Detecting morphed face images using facial landmarks. in International Conference on Image and Signal Processing. 2018. Springer.

[3.] Raghavendra, R., et al. Face morphing versus face averaging: Vulnerability and detection. in 2017 IEEE International Joint Conference on Biometrics (IJCB). 2017. IEEE.

[4.] Zhao, W., et al., Face recognition: A literature survey. ACM computing surveys (CSUR), 2003. 35(4): p. 399-458.

[5.] Turk, M. and A. Pentland, Eigenfaces for recognition. Journal of cognitive neuroscience, 1991. 3(1): p. 71-86.

[20.] Thomaz, C.E., R.Q. Feitosa, and A. Veiga. Design of radial basis function network as classifier in face recognition using eigenfaces. in Neural Networks, 1998. Proceedings. Vth Brazilian Symposium on. 1998. IEEE.

[21.] Sun, Y., et al. Deep learning face representation by joint identification-verification. in Advances in neural information processing systems. 2014.

[22.] Sun, Y., et al., Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873, 2015.

[23.] Sun, Y., X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[6.] Gottumukkal, R. and V.K. Asari, An improved face recognition technique based on modular PCA approach. Pattern Recognition Letters, 2004. 25(4): p. 429-436.

[7.] Yang, J., et al., Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE transactions on pattern analysis and machine intelligence, 2004. 26(1): p. 131-137.

[8.] Naseem, I., R. Togneri, and M. Bennamoun, Linear regression for face recognition. IEEE transactions on pattern analysis and machine intelligence, 2010. 32(11): p. 2106-2112.

[9.] Hu, H., P. Zhang, and F. De la Torre, Face recognition using enhanced linear discriminant analysis. IET computer vision, 2010. 4(3): p. 195-208.

[10.] Lu, J., K.N. Plataniotis, and A.N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms. IEEE Transactions on Neural Networks, 2003. 14(1): p. 117-126.

[11.] Yankun, Z. and L. Chongqing. Face recognition using kernel principal component analysis and genetic algorithms. in Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on. 2002. IEEE.

[12.] He, X., et al., Face recognition using laplacianfaces. IEEE transactions on pattern analysis and machine intelligence, 2005. 27(3): p. 328-340.

[13.] Ahonen, T., A. Hadid, and M. Pietikainen, Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2006(12): p. 2037-2041.

[14.] Zhang, B., et al., Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. IEEE Transactions on Image Processing, 2007. 16(1): p. 57-68.

[15.] Karczmarek, P., et al., An application of chain code-based local descriptor and its extension to face recognition. Pattern Recognition, 2017. 65: p. 26-34.

[16.] Nikan, S. and M. Ahmadi, Local gradient-based illumination invariant face recognition using local phase quantisation and multi-resolution local binary pattern fusion. IET image processing, 2014. 9(1): p. 12-21.

[17.] Zhang, L. and D. Samaras, Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. 28(3): p. 351-363.

[18.] Fleming, M.K. and G.W. Cottrell. Categorization of faces using unsupervised feature extraction. in Neural Networks, 1990., 1990 IJCNN International Joint Conference on. 1990. IEEE.

[19.] Lin, S.-H., S.-Y. Kung, and L.-J. Lin, Face recognition/detection by probabilistic decision-based neural network. IEEE transactions on neural networks, 1997. 8(1): p. 114-132.

[24.] Sun, Y., X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[25.] Taigman, Y., et al. Deepface: Closing the gap to human-level performance in face verification. in Proceedings of the IEEE

[26.] conference on computer vision and pattern recognition. 2014.

[27.] Taigman, Y., et al. Web-scale training for face identification. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[28.] Ren, S., et al. Face alignment at 3000 fps via regressing local binary features. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[29.] Best-Rowden L. Han H, Oto C, et al. Unconstrained face recognition: Identifying a person of interest from a media collection[J]. IEEE Transactions on Information Forensics and Security, 2014. 9 (12):2144-2157.

How to Cite This Article

Chamani, H., Nadjafi, M., Improved Reliable Deep Face Recognition Method Using Separated Components, Brilliant Engineering, 3(2022), 4563.
https://doi.org/10.36937/ben.2022.4563