DOCTORAL THESIS

Symmetrical Characteristic for Face Recognition

Graduate School of Science and Engineering, Iwate University
Design & Media Technology
Min Zou

March 2022

Acknowledgement

I would like to express my deepest gratitude to associate Prof. Takuya Akashi for his excellent guidance, patience, endless support, and for providing me a comfortable atmosphere for doing my research. I would like to thank Prof. Kouichi Konno, Prof. Takamitsu Tanaka, who gave me many useful comments and advice since I started my research.

I would like to thank every member of the Smart Computer Vision Lab. of Iwate University for their help and patience. I would especially like to thank Dr. Yi Zhang for his help and advice. I also would thank my husband, Dr. Mengbo You and my daughter Mizuki You, who accompanies me, facing many difficulties and sharing all the happiness together. At least, I would like to thank my parents, who always support me in pursuing my dream without asking for any return. I can never forget that they gave all they have to support me to start my study in Japan.

Abstract

Face recognition refers to the technology capable of identifying or verifying the identity of subjects in images or videos. Face recognition has a wide range of applications, which can be used in automatic access control systems, identification of ID cards, and home security. Facial recognition systems can identify individuals by comparing the input image to the stored or learned images. Based on the fact that different individuals have different facial features, face recognition systems usually take advantage of analyzing the characteristics of each individual's face. Compared with the common characteristics of many individuals, the unique characteristics are usually emphasized to distinguish one identity from another. Since the first face recognition algorithms developed in the early seventies, the accuracy of face recognition improved to the new level that nowadays face recognition often performs more practical and more convenient than any other biometric modalities that have traditionally been considered more robust, such as fingerprint recognition and handwriting recognition.

Most face detection and recognition tasks are based on the training of intact facial images and corresponding labels. Both the three-dimensional structure and two-dimensional appearance from the frontal view of human faces are approximately bilaterally symmetrical in general. However, sometimes, illumination on

the left-half face and the right-half face is uneven. In this case, the symmetrical characteristic of human faces can facilitate expressing distinct identity information. This is because even if one side of the facial image is corrupted by noise, the opposite side can still be used for feature extraction. The recent literature indicates that face recognition and facial expression classification has achieved high accuracy on benchmark datasets with a large number of face images in the wild. However, unlike the purpose of recognizing as many people as possible, real applications for families or companies usually aim to recognize a small group of people as accurately as possible. In case of the face is partially occluded, convolutional solutions always simply put images with occlusions into the training dataset and hope the convolution neural network learns a model robust to partial occlusion. These processes not only increase the burden of learning but also affect the model to identify normal images without occlusions.

To address this problem, an automatic selection of the better half of the face can be used for identity recognition with only a single half face. Different from the MegaFace challenge of recognizing millions of identities in the wild, this thesis focuses on building recognition systems for a small number of people with fewer training images, for example, building access control systems for research laboratory members or family members. This thesis proposes an artificial face image construction method and a half-face training strategy for transfer learning of pre-trained conventional neural network models.

The facial image reconstruction to discard the influence of partial occlusion is also discussed. Based on the phenomenon that human faces are roughly symmetrical, the intact half-face can be used to reconstruct the facial information of the

occluded areas. Specifically, occlusion on the left-half face is reconstructed with a linear combination of features on the right-half face and vice versa. The process is modeled by keeping row sparsity for the coefficient matrix with l2,1-norm regularization while minimizing the reconstruction error. An alternative iterative algorithm is proposed to solve the optimization problem. To validate the effectiveness of the reconstruction, the pre-trained CNN model is trained on normal face images and tested with various occluded images. Extensive experimental results show that the proposed method improves the performance of state-of-the-art models by utilizing the symmetrical characteristics of human faces.

Contents

Chapter 1

Introduction

1.1 Background

Face recognition refers to the technology capable of identifying or verifying the identity of subjects in images or videos [1, 2, 3]. Since the first face recognition algorithms [4, 5] developed in the early seventies, the accuracy of face recognition improved to the new level that nowadays face recognition often performs more practical and more convenient than any other biometric modalities that have traditionally been considered more robust, such as fingerprint recognition and handwriting recognition [6, 7, 8].

A significant difference between face recognition and any other biometric modalities is that it has many influence factors, such as illumination, expression, pose, occlusion, and so on. The limited facial features will be affected by these factors. These factors will cause a significant intra-class difference, which may be greater than the inter-class difference. This is contrary to the objective of face recognition, i.e., maximizing the discriminating power between inter-class information and minimizing the differences between intra-class information [9].

In the first decade of the 21st century, with the development of machine learning, researchers have successively explored face recognition method which based on genetic algorithm [10], support vector machine (SVM) [11], boosting [12] and manifold learning [13]. Then, sparse representation [14] became a research hotspot because of its robustness to occlusion factors. At the same time, the industry has basically reached a consensus: feature extraction based on artificially designed local descriptors and subspace methods for feature selection can achieve the best recognition results. Gabor and LBP feature descriptors [15] are two of the most successful artificially designed local descriptors in the field of face recognition. During this period, the targeted processing of various face recognition influence factors is also the research hotspot of that stage, such as face illumination normalization, face pose correction, face super-resolution, and occlusion processing. Also at this time, the research about face recognition was moved from a constrained environment to an unconstrained environment. The LFW [16] Face Recognition Open Competition became popular in this context. The best recognition system at that time can achieve more than 99% recognition accuracy on the FRGC dataset which is constrained.

In 2013, researchers at Microsoft Research Asia first attempted large-scale training data of 100,000 scales and obtained 95.17% accuracy on LFW based on high-dimensional LBP features and the joint formulation [17]. This result indicates that large training data sets are important for effectively improving the accuracy of face recognition in an unconstrained environment. However, all of these classic methods are difficult to handle training scenarios for large datasets.

Around 2014, with the development of big data and deep learning, neural networks have attracted more and more attention, and have obtained far more results than classical methods in image classification, handwriting recognition, and speech recognition. More and more researchers apply neural network to face recognition, and getting higher and higher accuracy [18, 19, 20, 21, 22].

Since then, researchers have continued to improve the network structure, while expanding the scale of training data, pushing the recognition accuracy on the LFW [16] to more than 99.5%. The basic trends based on some classic methods in the development of face recognition is as follows: the scale of training data is getting larger and larger, and the recognition accuracy is getting higher and higher.

1.2 Research Problem

Face recognition is a typical computer vision task based on classifying facial image features. In particular, because deep learning algorithms achieve good performance in image classification, numerous convolutional neural network (CNN)-based methods have been proposed to recognize millions of human identities in large face image datasets. This is reasonable considering that facial analysis is based on the extraction and comparison of key features of the face. Among them, the integrity of the feature is the key to the success or failure of a face recognition algorithm [23]. In the case of extracting image features, once some parts of the features disappear owing to the occlusion or uneven illumination of the face, it will lead to fail or unsatisfied results. It will also be impossible to compare with the face information in the database. The difficulty of face recognition caused by occlusion or uneven illumination is mainly reflected in the feature loss, alignment error, and local aliasing [24].

In some cases, the face recognition of several people requires only a few images. For instance, when we wish to build an access control system for research laboratory members or family members using face recognition techniques, the ability to recognize millions of human identities is not necessary. In this case, recognizing a few people is more important than recognizing millions of people. In addition, collecting millions of images is difficult and time-consuming. Therefore, this thesis focuses on the task of recognizing a few people with a small number of training images for practical use.

In small datasets, facial feature extraction may be affected by factors such as illumination, expression, pose, and occlusion. These factors will cause significant facial feature information intact. This thesis proposes a more effective extraction of features using the symmetrical characteristics of human faces to reduce the effects of these factors, particularly the uneven illumination conditions and occlusion on the left and right half faces, as shown in Fig. 1.1 and Fig. 1.2, especially the left half face or the right half face is totally occluded, as shown in Fig. 1.3. As illustrated in the comparison between Fig. 1.4 and Fig. 1.5, the method proposed in this thesis is able to address the classification problem in all the cases that are shown in Fig. 1.1, Fig. 1.2, and Fig. 1.3 [25] [26]. For the uneven brightness on the face in the face recognition task, there are also other researches, such as lighting normalization method. The underlying reflectance model is proposed in [27]. This method characterizes interactions between skin surface, lighting source and camera sensor, and elaborates the formation of face color appearance. The experimental results show that it is effective by improving the face recognition task. However, this kind of methods is complex and time-consuming. So, this thesis propose a simple flipping method to handle with the left-right uneven illumination problem and left-right occlusion problem.

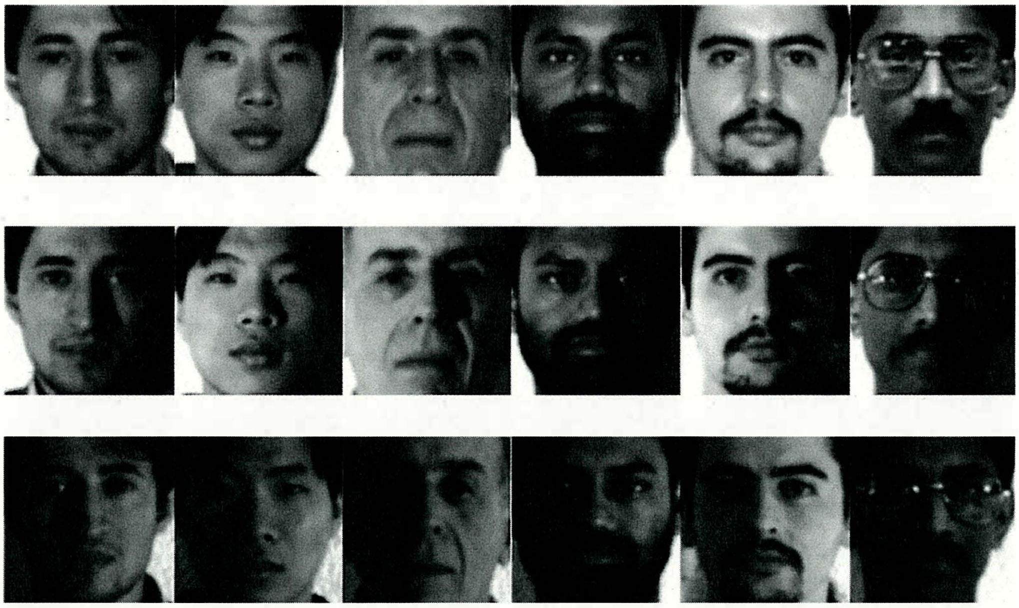


Figure 1.1: Sample images of uneven illumination on left-half and right-half of face. The first row shows normal illuminated images. The second row shows images with a light source located on the left. The third row shows images with a light source located on the right.

Considering that different individuals have different facial image features, face recognition systems typically analyze the distinctive characteristics of each individual's face. A face recognition system typically requires a large number of images for each individual to assemble the training image dataset. Most publicly accessible large face image datasets contain images captured in the wild and from different views, thereby resulting in many profile face images. Some images are collected directly from the Internet with unknown copyright problems. However, when building face recognition systems for a small number of people, it is difficult to collect the same number of images as the number of images in public datasets. When only a

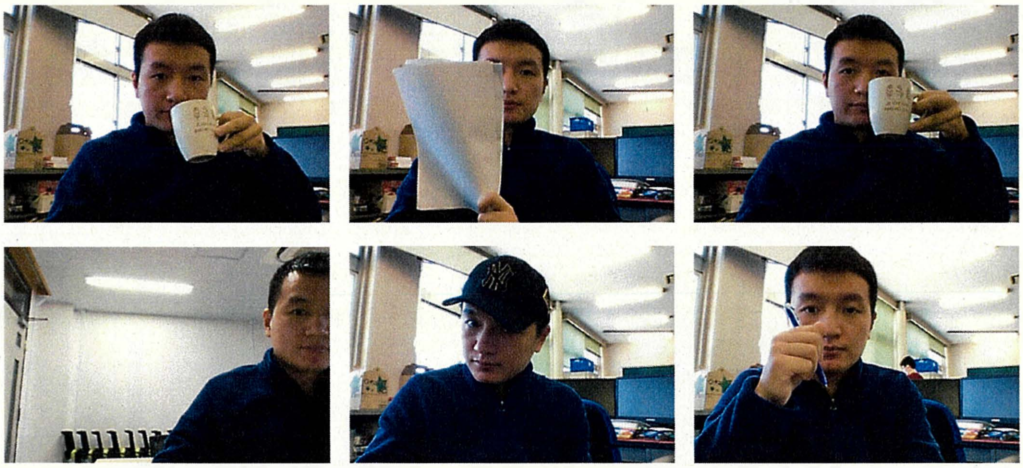


Figure 1.2: Examples of occluded faces.

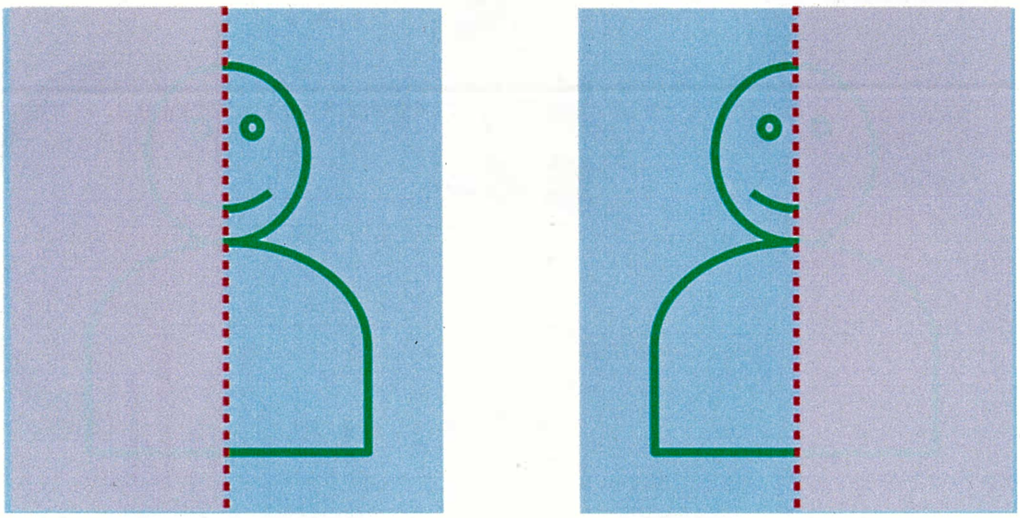


Figure 1.3: The cases of left half face occluded and right half face occluded.

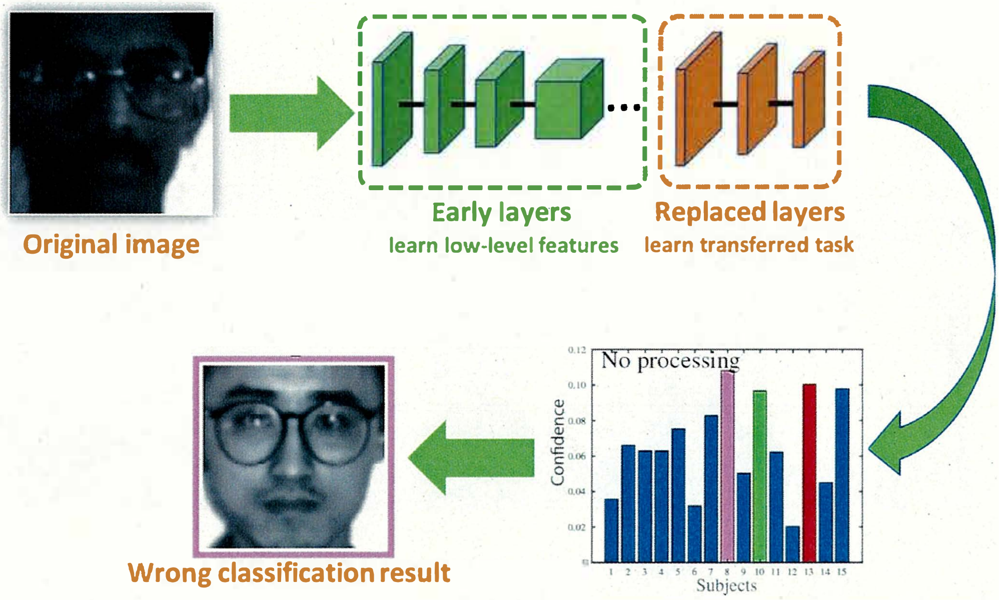


Figure 1.4: The case of applying the existing transfer learning method.

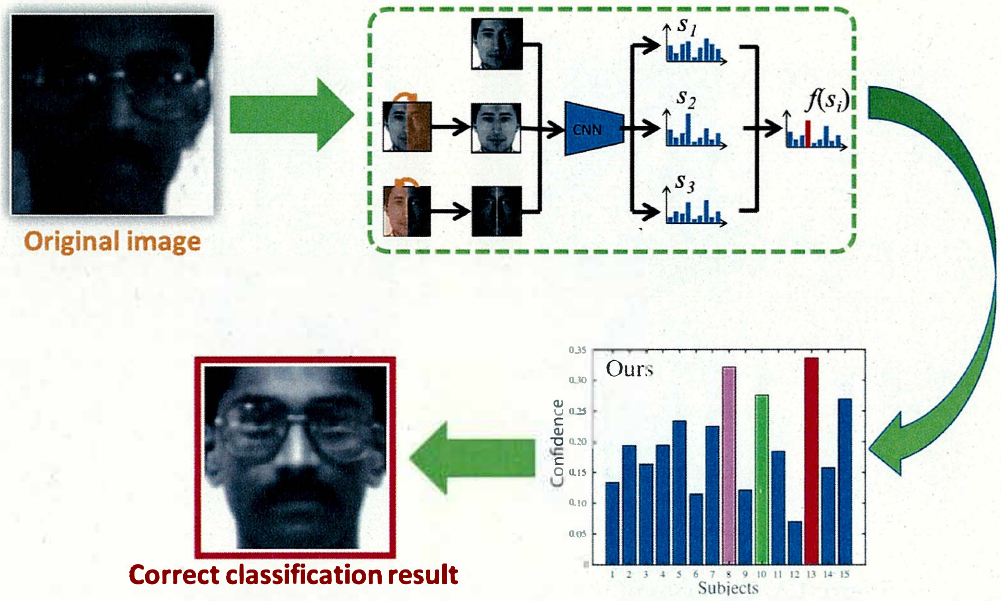


Figure 1.5: The case of applying the proposed method.

few images for each individual can be used, transfer learning of pre-trained CNN models on ImageNet is a good alternative for reusing the parameters trained for object classification.

In the past two decades, feature extraction with artificially designed local descriptors and subspace methods for classification have been investigated successively based on particle swarm optimization, support vector machine, boosting, and manifold learning. Subsequently, sparse representation [14] demonstrated robustness to occlusion. Significant attention has been directed to improving the performance of classification, which is affected by various factors, such as face illumination normalization, face pose correction, face super-resolution, and occlusion processing. Furthermore, studies regarding face recognition have evolved from a constrained environment to an unconstrained environment. With the development of big data and deep learning, CNNs have attracted significant attention and achieved better results than existing methods in many image classification tasks. Moreover, face recognition methods using CNNs have achieved high accuracies in many public datasets [18, 19, 20]. Since then, the scale of training data for CNNs has increased significantly, and the recognition accuracy has improved considerably. When training image shortage occurs, pre-trained networks from other large datasets can be reused by transferring the network for new classification tasks [28]. Hence, both the number of images required for training and the training time can be reduced significantly by transfer learning. This thesis proposes the application of facial symmetrical characteristics to transfer learning.

Owing to the effects of varying illumination conditions on face recognition, Pizer et al. [29] proposed histogram equalization to normalize faces with different illumination effects. Shan et al. [30] developed a gamma intensity correction to normalize the overall image intensity at a specified illumination level; it performed well in different situations. Blanz and Vetter [31] proposed a face recognition method suitable for a three-dimensional (3D) deformation model that describes the shape and texture of the face separately. This model achieved good results on the CMU-PIE [32] and FERET [33] datasets. Ishiyama et al. [34, 35] established a 3D shape data model under arbitrary illumination and attitude, fitted the model in certain lighting conditions and posture positions, and then assessed the results. In a 200-individual dataset, different illumination datasets, and multi-pose image sets (a total of 14,000 datasets), the average recognition accuracy rate reached 93.8%.

In the field of hand-crafted features, partial occlusion has become an important challenge for facial analysis, especially in face recognition. It has received widespread attention. Prior to the use of CNNs, the methods for partial occlusion were mainly divided into two categories. One was to extract only the part of the face that was not occluded, and the other was to recover the complete normal non-occluded face from the occluded part [36].

In order to handle the occlusion problem in face recognition, Kim et al. [37] proposed a method based on two-dimensional principal component analysis to combine the k-nearest neighbors and 1-nearest neighbors classifier to eliminate the occlusion effect and then do the partial matching only on the non-occluded parts. Their results showed that their method was robust to occlusions by sunglasses or scarfs. Another literature [38] introduced a support vector machine (SVM) based approach for partially occluded face recognition. When there is an occlusion in the training or test datasets, the feature vector of the defined sample will lack entries and the conventional SVM will not be able to handle the problem. They defined a criterion that minimizes the probability of overlap to solve this partial occlusion problem. To indicate the possible range of values for missing entries, the derivation algorithm incorporates additional terms into SVM to yield better classification results. However, it still remains a challenge to discard the influence of occlusion while recognizing the identity with faces.

This thesis proposes a more effective extraction of features using the symmetrical characteristics of human faces to reduce the effects of the affect factors, such as illumination and occlusion. In [26], occlusion on the left-half face is reconstructed with a linear combination of features on the right-half face, and vice versa. The process is modeled by keeping row sparsity for the coefficient matrix with $l_{2,1}$-norm regularization while minimizing the reconstruction error. This paper showed that the occluded half face can be reconstructed by unoccluded half. This means that the symmetrical characteristic of human faces is effective to perform the face recognition task. In [25], an automatic selection method of the better half of the face using only a half-face for identity recognition is proposed. This paper proposes an artificial face image construction method and a half-face training strategy for transfer learning of pretrained conventional neural network models. Extensive experimental results show that the proposed method improves the performance of state-of-the-art models by utilizing the symmetrical characteristics of human faces.

Chapter 2

Related Works

In various domains, face recognition has many applications, such as face recognition attendance system, face recognition anti-theft door, and so on. By identifying or verifying a person in video frames through the facial biometric pattern and data, face recognition has a wide range of applications in life. Therefore, face recognition received a great deal of attention over the last few years.

As a biometric technique, face recognition has several advantages. Firstly, facial recognition technology is based on facial photos or real-time facial images, which is undoubtedly the easiest to obtain compared with iris recognition, fingerprint scanning, palm scanning, and other technologies. Low cost, easy to promote and use. Since face recognition technology uses conventional general-purpose equipment, the price is within the acceptable range of general users. Compared with other biological recognition technologies, face recognition products have a high cost-performance ratio. The equipment used in face recognition technology is ordinary PCs, cameras, and other conventional equipment. Since computers and closed-circuit television monitoring systems have been widely used, most users do not need to purchase a large number of special equipment to use face recognition technology. This not only protects the user's original investment, but also expands the functions of the user's existing equipment, and meets the user's security requirements. Secondly, in applications with high-security requirements, face recognition technology requires that the recognition object must be at the recognition site in person, and it is difficult for others to counterfeit. The unique active discrimination ability of face recognition technology ensures that others cannot deceive the recognition system with inactive photos, puppets, and wax figures. This is difficult to achieve with biometric technology such as fingerprints. Thirdly, Face recognition technology uses a general-purpose camera as an identification information acquisition device to complete the recognition process in a non-contact manner without the recognition object being noticed. Intuitiveness highlights the use of facial recognition technology based on human facial images, and human faces are undoubtedly the most intuitive source of information that can be distinguished by the naked eye, which is convenient for manual confirmation and auditing. "Judge people by appearance" conforms to the law of human cognition.

However, face recognition also has some disadvantages, it is not a perfect modality for all users. First, the facial feature information is unstable, which will change significantly over time due to aging, substantial changes in weight, lifestyle conditions, and modifications such as cosmetic surgery. Furthermore, face recognition is also affected by illumination conditions (such as day and night, indoor and outdoor), occlusion(such as masks, sunglasses, hair, beards, etc), and other unconstrained factors. Finally, since faces are one of the most prominent identifiers in modern society, face recognition meets the problems of the acceptability criterion, concerns about privacy [39, 40, 41].

Based on the images or videos obtained from surveillance systems, private cameras, or other hardware, a 2D face recognition system can be built in the online mode or offline mode. For the automatic face recognition system, there are always three steps. Firstly, the system has to detect the face in the input images or videos and segment it from the detected area. Then, some predefined canonical structure should be aligned. And these methods must treated to account for potential illumination changes. Finally, extract the facial features from the aligned images. Based on the calculated features, identity recognition is performed using a proper classification approach.

2.1 Holistic Methods

There are three different 2D face recognition method, (1) holistic methods, (2) geometrical methods, and (3) deep learning-based methods [42]. The holistic methods use the entire face region as input. Principal component analysis(PCA), known as eigenfaces [43] is the linear technique employed for facial recognition systems. Eigenfaces represent the main components of facial distribution. To overcome the problem of performance degradation due to illumination variability, based on the PCA methods, Zhao and Yang [44] presented a method for calculating the covariance matrix employing three images acquired under different light conditions to account for random lighting effects when the subject is Lambertian. The holistic methods for face recognition are prevalently used for face recognition systems in the 20th century. However, it is very sensitive to context changes and misalignments. In the majority of cases, the face must be cut manually from the image. Moreover, it is necessary to enforce geometric consistency in all facial instances because the data set is viewed as a single matrix. All of the facial images must be carefully matched within a standard frame of reference. In the face orientation, a minor error can cause substantial facial classification errors.

2.2 Geometrical Methods

Geometrical methods use facial landmarks to do face recognition tasks. Based on the geometric distribution, the landmarks of the face can be used to register facial features, the normalization of expressions, and the recognition of defined positions. Frank Y.Shih and Chao-Fa Chuang [45] represented a geometric face model to locate facial features and an elliptic model to trace face boundary to overcome the noises or clustered facial features candidates. Kumar et al. [46] represented an ensemble face recognition system that makes use of a local descriptor called Dense Local Structure. It uses an additional graph structure which is generated by finding additional corner pixel points through bilinear interpolation of neighborhood pixels. This method showed good performance in both constrained and unconstrained environments. For the geometric-based face recognition method, all facial images must be aligned to possess all referential points which contain mouth, nose, and eyes. It is usually considered as a challenging task that optimal automatic alignment.

2.3 Deep Learning Methods

For 2D face recognition, deep learning methods have become the dominant approach. Convolutional Neural Network (CNN) is the most commonly used deep learning method in face recognition. The main advantage of the deep learning method is that a large amount of data can be used for training. The deep learning model for face recognition can learn the face feature that is robust to changes in the training data. This method does not need to design specific features that are robust to different types of intra-class differences (such as lighting, posture, facial expressions, age, etc.), but can learn them from training data. The main shortcoming of deep learning methods is that very large datasets for training are necessary, and these datasets need to contain enough changes so that they can be generalized to unseen samples. Fortunately, some large-scale face datasets containing natural face images have been published and can be used to train CNN models. In addition to learning discriminative features, neural networks can also reduce dimensionality, and can be trained into classifiers. CNN is considered to be an end-to-end trainable system and does not need to be combined with any other specific methods.

With the development of big data and computing systems, deep-learning-based methods for face recognition have become increasingly popular. CNNs are considered end-to-end trainable systems, which can not only learn discriminative features, but also reduce dimensionality. Furthermore, image classification classifiers can be implemented after the training process in CNNs. The ILSVRC competition [47] involved a task classifying an image into one of thousands of categories, where AlexNet [48] achieved remarkable classification results. AlexNet deepens the structure of the network based on LeNet and learns features that are richer and have more dimensions. Subsequently, GoogLeNet [49] became more attractive because of its simple structure and good performance in the classification task. With the development of CNNs, an increasing number of models have been developed in the field of face recognition, such as DeepFace [19], FaceNet [50], and CosFace [51]. These methods yielded successful models for face recognition with a complicated CNN structure. A large-scale training dataset was required when using these models. When we cannot collect too many images instantly and only need to classify a small group of people, eg. research lab members, transfer learning with these models is a proper choice.

However, the aforementioned methods are based on images of the entire face. Takano [52] mentioned that the human body is approximately left–right symmetric and people may sometimes be confused of the left and right when viewing a mirrored image. As part of the human body, the face exhibits this left–right symmetry. Hence, the similarity of the left- and right-half face features can be used for face detection. In 2014, You et al. [53] proposed the use of facial symmetry characteristics for profile face detection. In 2016, Xu et al. [54] used symmetry to perform face image preprocessing and virtual face image data augmentation. In the research field of face recognition, recent studies have focused on a larger number of images in the dataset and better CNN structures. However, few studies have focused on the characteristics of the face itself for simplifying face classification. This thesis proposes a method to deal with the left-right uneven illumination on face utilizing the symmetrical characteristics of human faces and a method to reconstruct the occluded half face by the unoccluded half.

Chapter 3

Facial Symmetrical Characteristic

3.1 Flipping Half Face for Testing

Face recognition is sensitive to many external factors, such as illumination, make-up, and occlusion. These factors can confuse the identification of a person. For example, under varying lighting conditions, it is challenging to identify a person correctly. In fact, it has been argued that two images of the same person are less similar than two images of different persons based on the change in illumination [55]. This thesis proposes the use of symmetrical characteristics to improve the similarity between two images of the same person and reduce the effect of noise.

3.1.1 Flipping Strategy

This idea is motivated by the typical phenomenon of mirror reversal. When we stand in front of a mirror, we may easily confuse the left-half face with the right-half face. The human body is completely different from the top to the bottom or from the front to the back; however, by mirroring, the similarity of the left- and

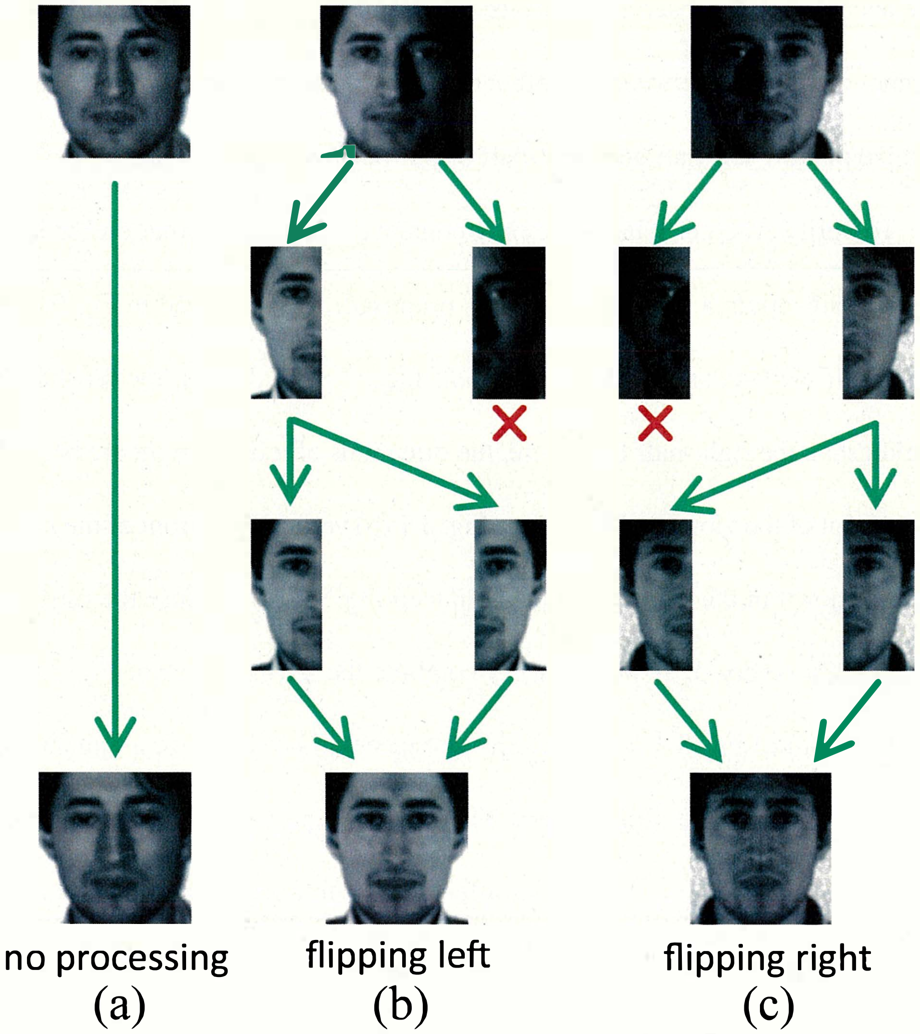


Figure 3.1: Flipping Scheme. (a) A face that is unaffected by illumination and does not require processing. (b) A face where the left-half is lighter than the right-half. The left-half face that is not affected by illumination is flipped to the right and then combined with the original left-half face to form a conjecture face. (c) A face where right-half is lighter than left-half. The right-half face that is not affected by illumination is flipped to the left and then combined with the original right face to form a conjecture face.

right-half body can be recognized owing to the left-right symmetry of the human body. As part of the human body, the face exhibits this left-right symmetry. People are accustomed to observing our faces using mirrors or photographs. Although we cannot distinguish between the left and right parts of our body through a mirror, we realize that the left-half and right-half of our body are approximately symmetrical.

To verify whether a half-face may contain sufficient information for facial image classification, a flipping strategy is proposed. As illustrated in Fig. 3.1 (b), the right-half face is darker than the left-half face. In Fig. 3.1 (c), the left-half face is darker than the right-half face (here, the directions of left and right are based on the viewpoint of the viewer). If we place Fig. 3.1 (b) vertically in front of the mirror, the image shown in the mirror will be similar to Fig. 3.1 (c) because the mirror image is left-right reversed. If we manually combine the left-half face of Fig. 3.1 (b) and the right half of Fig. 3.1 (c), the conjecture face will be similar to a human face that is not affected by the illumination. Therefore, we performed a left-right mirrored rotation of the half-face that is not affected by illumination and then combined it with the original half-face to generate a conjecture face. This process will reduce the effect of illumination on face recognition accuracy.

Evidence was provided via mirror reversal as follows:

- We are accustomed to observing our face via mirrors or photographs. Although we cannot distinguish between the left and right parts of our body in the mirror, we realize that the left-half and right-half of our body are approximately symmetrical.

- For face recognition, we can flexibly use the faces that are reconstructed by the left or right half-face.

- Because we used the post-processing method to flip the face image, this flipping process will not affect the performance of the classifier.

3.1.2 Symmetry Extraction

The cropped frontal face images exclude most background regions. However, the symmetry line is not always located in the middle line of the image. In most cases, the symmetry line is slightly shifted from the middle line of the image. Because we aim to obtain the flipping face, the first step is to identify the symmetry line of the face. To obtain an accurate symmetry, a textual- and color-histogram-based symmetry detection method [56] was applied. However, in our case, the prior information that the symmetry is close to the middle line and almost vertical can be used to simplify the symmetry detection. First, we applied a simplified log-Gabor filter to calculate the image response. The applied two-dimensional (2D) log-Gabor filter (written in polar coordinates) in the frequency domain is defined as follows:

$$\begin{aligned} G(\omega,\theta) =& G(\omega)G(\theta) = \\ & \exp\left\{\frac{-\left[\ln(\omega/\omega_0)\right]^2}{2\left[\ln(k/\omega_0)\right]^2}\right\}\exp\left\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\right\}, \end{aligned} \tag{3.1}$$

where ω_0 denotes the central frequency, k denotes the radial bandwidth, θ_0 denotes the orientation, and σ_θ denotes the angular bandwidth. To obtain filters of the same shape, k/ω_0 was fixed in our case. Subsequently, the face image was transformed using the log-Gabor filter. In our case, four scales and five orientations were applied to extract the features. The maximum amplitude was used as the response image.

Finally, the textural- and color-histogram-based symmetry triangulations were the same as those proposed in 2017 [56]. As shown in Fig. 3.2, the symmetry extraction result is shown on the left, whereas the corresponding probability map is displayed on the right. Most extracted symmetries are slightly shifted from the middle of the image. Therefore, flipping will enable the symmetry to be the same as the extracted symmetry; consequently, more accurate flipped images can be obtained.

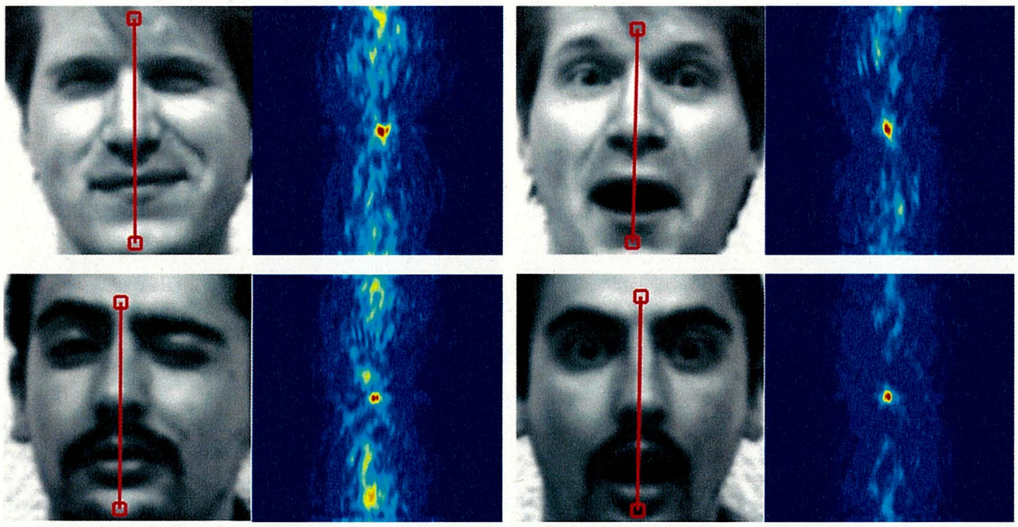


Figure 3.2: Symmetry extraction results.

3.1.3 Symmetry Adjustment

To flip the face along an inclined symmetry, an in-plane rotation was applied to the image, as shown in Fig. 3.3. Suppose an arbitrary pixel at (x, y) in image I with width w and height h is rotated with angle α to be (x', y'). The correspondence can be calculated using the following trigonometric function:

$$\begin{cases} x' = \left(x - \frac{w}{2}\right)\cos\alpha - \left(y - \frac{h}{2}\right)\sin\alpha + \frac{w'}{2}, \\ y' = \left(x - \frac{w}{2}\right)\sin\alpha + \left(y - \frac{h}{2}\right)\cos\alpha + \frac{h'}{2} \end{cases} \tag{3.2}$$

Because the rotation is around the image center, the center of the symmetry line is first shifted to the image center. Strictly cropped images of the Yale dataset only require a slight rotation, and images captured in the wild require a large rotation.

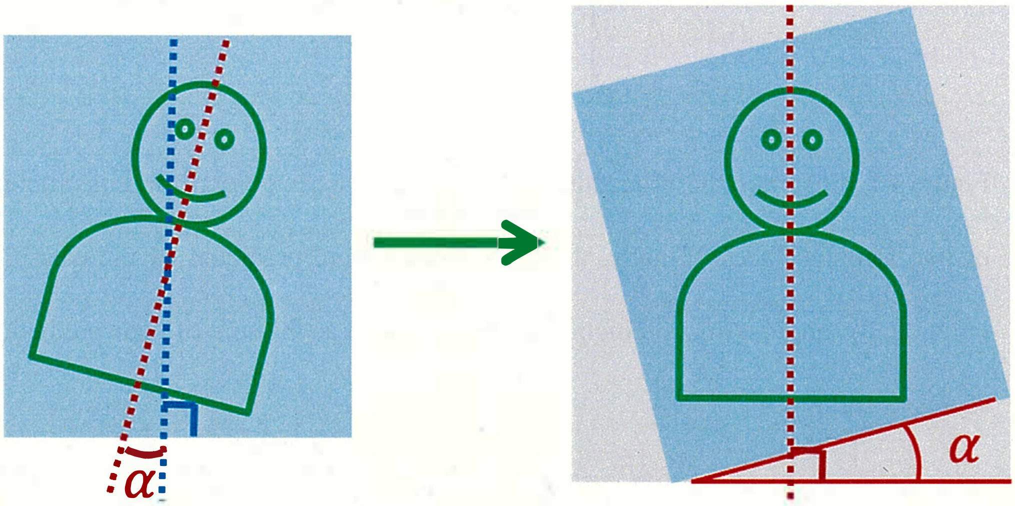


Figure 3.3: Rotate the image with angle α to make the symmetry line vertical.

3.1.4 Flipping Scheme for Classification

The reconstructed face can be classified into two cases according to the flip direction. In the first step, the strictly cropped image of one face is segregated into two sections of the same area on the left and right. Let $I(x, y)$ denote an arbitrary pixel in image I with width w. In one case, flipping the left face creates a new image I_L as follows:

$$I_L\left(x, y\right) = \begin{cases} I\left(x, y\right), & 0 \leq x \leq \frac{w}{2} \\ I\left(w - x, y\right), & \frac{w}{2} < x < w \end{cases} \tag{3.3}$$

The candidate area is the left-half face. The left-half face was flipped to the right side and combined with the original candidate area to construct a conjecture whole face. In another case, the candidate region is the right-half face. Hence, the right-half face was flipped to the left and combined with the original candidate region to construct a conjecture whole face I_R as follows:

$$I_R\left(x,y\right)=\begin{cases} I\left(x-\frac{w}{2},y\right), & 0\leq x\leq \frac{w}{2} \\ I\left(x,y\right), & \frac{w}{2}<x<w \end{cases} \tag{3.4}$$

Figure 3.1 illustrates the procedures of the flipping scheme and the construction of two artificial faces. Finally, the three images were predicted by the classifier individually, and the final prediction score $\hat{p}(I)$ is set as the sum of their prediction scores as follows:

$$\hat{p}\left(I\right)=p\left(I\right)+p\left(I_L\right)+p\left(I_R\right) \tag{3.5}$$

Because the prediction scores will be used to find the maximum one, and the identity with the maximum prediction score is supposed to be the recognition result, the average of three prediction scores is equivalent to the sum of them. Calculating the sum of three prediction scores is also faster than the average.

Using the flipping scheme method, we can not only solve the effect of uneven illumination on the left and right half-faces but also perform face recognition smoothly. Compared with existing methods, flipping half face for testing constructs three new images and obtains three corresponding confidence vectors which needs to be summarized to output the final decision, as shown in Fig. 3.4. In recent years, many face recognition methods for illumination primarily train deep learning models to learn many images affected by illumination. The premise of this method is to collect rich data affected by illumination. However, the proposed method solves this problem from another perspective, where a significant amount of data affected

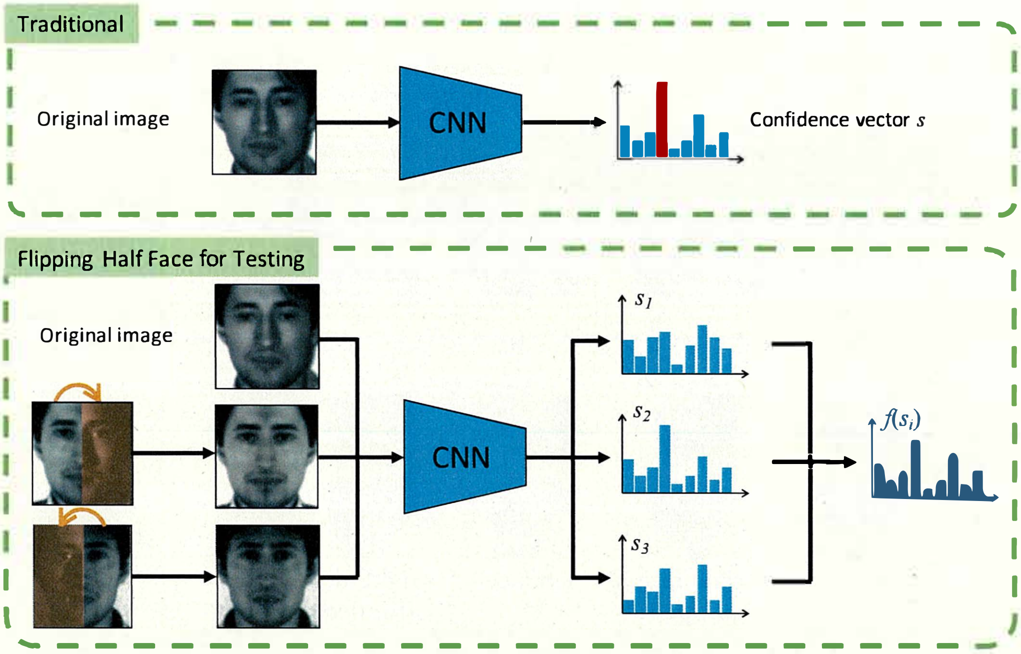


Figure 3.4: Comparison between traditional methods and flipping half face for testing.

by illumination need not be learned in advance, and recognition can be performed by the conjectured face using a flipping scheme as the post-processing method after learning the normal face images.

3.2 Training with Half Faces

In the previous section, flipping a half-face was proposed to construct an artificial face image for testing, i.e., a method that did not change the normal training process. However, the artificial face image incurred new noise around the symmetry, rendering it visually different from normal faces. Therefore, we propose another method to apply facial symmetrical characteristics. In this method, all the whole-face images are cut into halves, which can be denoted as left- and right-faces. To maintain an organized dataset, the right faces were mirrored such that they appear similar to the left faces, as shown in Fig. 3.5. Subsequently, the six CNN models

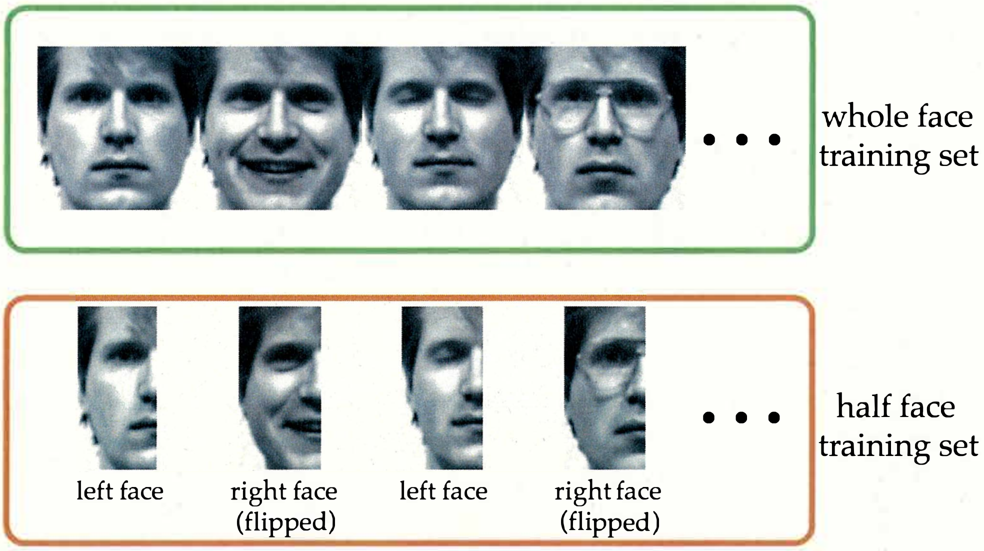


Figure 3.5: Illustration of whole-face training set and half-face training set. The left-half or right-half for each face was selected randomly, and the right-half was flipped.

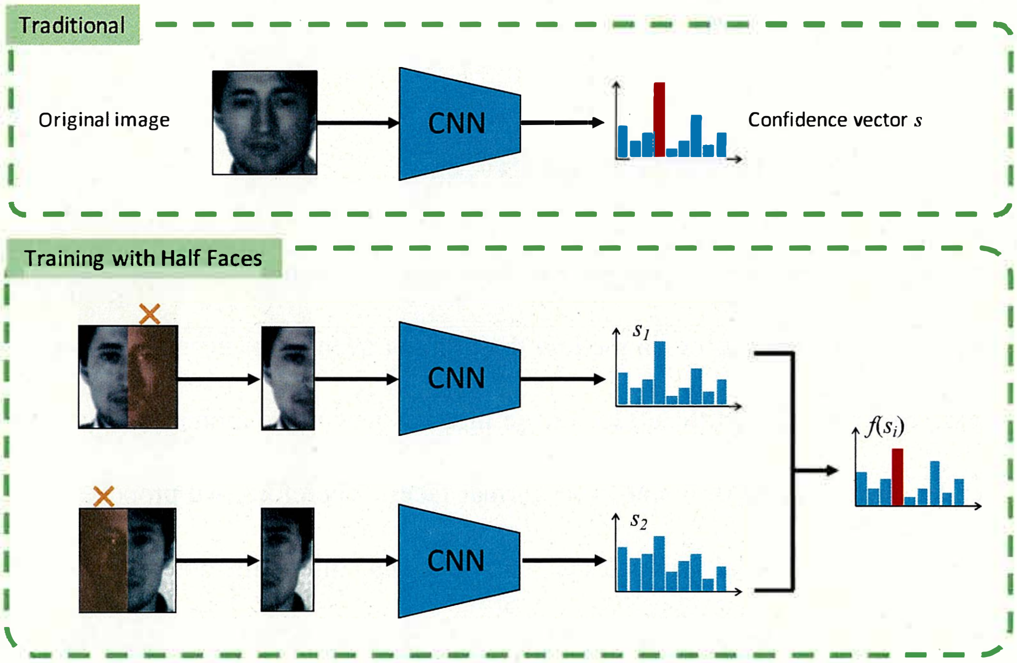


Figure 3.6: Comparison between traditional methods and training with half faces.

were trained on the new training images. In the testing stage, both the left-half and the mirrored right-half of the whole-face testing images were input into the classifier, and the half-face with a higher prediction score was used as the final decision. We did not only use the left face to perform the training owing to two reasons. First, as each whole face was cut into two half-face images, the number of images would be doubled. For a fair comparison, we randomly selected only half of the whole face and did not use the other half. Second, in practical applications, the left face might be poorly illuminated, and hence the right face would be more useful for the identification task. This training strategy will nullify the mutual effect between the two halves, and learning will be emphasized from each half-face image. Compared with existing methods, training with half faces splits the whole face image into halves and obtains two confidence vectors for each of them, which needs to be summarized to output the final decision, as shown in Fig. 3.6.

3.3 Face Reconstruction

3.3.1 Problem Formulation

For partial occlusions or uneven illumination which only covers an area smaller than half of the whole face image, there is a large possibility that the face is only half occluded or uneven illuminated while the other half face normal. In this case, the facial information in the occluded or uneven illuminated area is corrupted and cannot be used for facial analysis. However, the facial information outside of the occluded area or uneven illuminated area remains the same as the raw data. We proposed a novel method to reconstruct the corrupted facial information of the occluded region with uncorrupted dimensions of the raw data.

Considering that the frontal view faces are usually roughly symmetrical along the middle vertical line, it can be assumed that the left-half and the right-half of frontal view human faces carry almost the same information but distributed with mirror reversal [57, 58]. When a small part of the human face is occluded or the illumination is not good, people tend to imagine the occluded part or dark part with other normal parts [54]. In this way, small occlusion or dark part does not affect the facial information analysis of our brain. Motivated by the visual information processing mechanism, we assume that the corrupted facial feature on either half-face can be reconstructed with features on the other half face. Specifically, each dimension of the facial features can be reconstructed by a linear combination of features on the other half face. A representative feature that is close to the symmetrical position of the corrupted feature should give a large weight in the combination.

Suppose there is a dataset containing n facial images, the left-half parts of all images are denoted by $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the right-half parts of all images are denoted by $\widetilde{\mathbf{X}} \in \mathbb{R}^{n \times d}$, as shown in Fig. 3.7 and Fig. 3.8. Each half of an image is reshaped to be a vector of d dimensions. Each dimension of all the left-half and right-half images in the dataset can be represented by $\mathbf{f}_i$ and $\widetilde{\mathbf{f}}_i$ respectively. For each feature vector $\mathbf{f}_i$, the corresponding reconstruction optimization problem can be written as:

$$\min \sum_{j=1}^{d} |w_{ji}|_p, \quad \text{s.t. } \mathbf{f}_i = \sum_{j=1}^{d} w_{ji}\widetilde{\mathbf{f}}_j, \tag{3.6}$$

where w_{ji} represents the reconstruction coefficient and $|.|_p$ denotes the p-norm. The

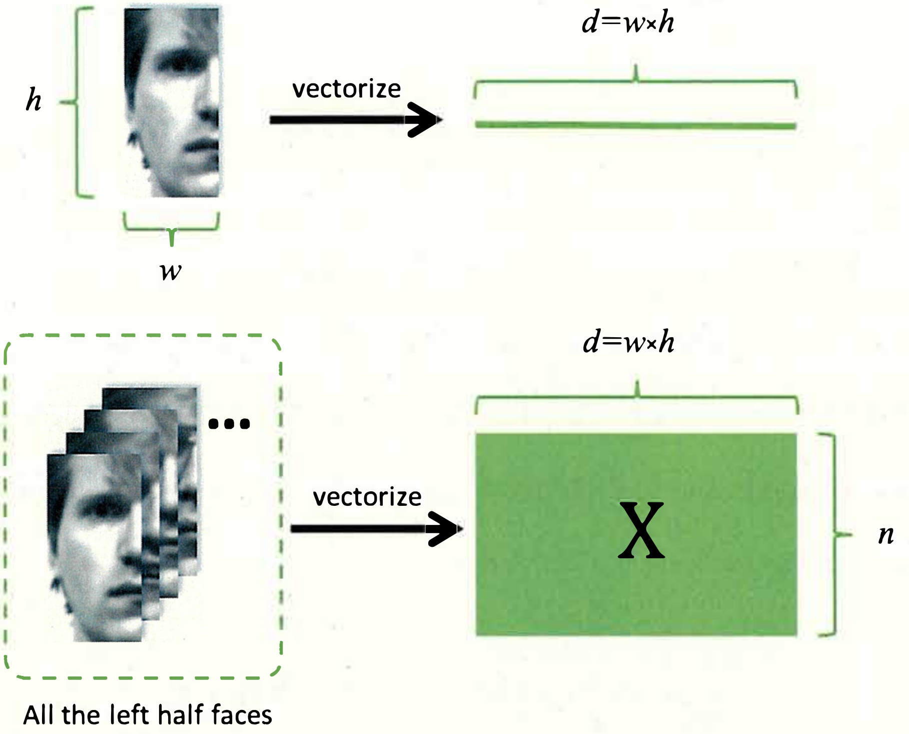


Figure 3.7: The process to construct matrix **X**.

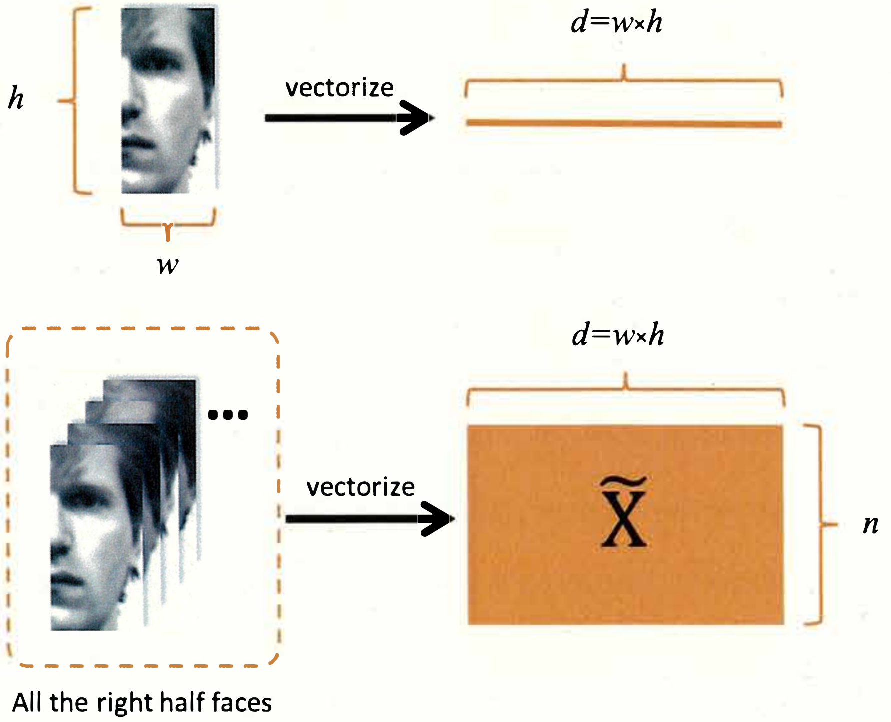


Figure 3.8: The process to construct matrix $\tilde{\mathbf{X}}$.

corresponding matrix form can be formulated as:

$$\min_{\mathbf{W}} ||\mathbf{W}||_p, \text{ s.t. } \mathbf{X} = \widetilde{\mathbf{X}}\mathbf{W}, \tag{3.7}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the reconstruction coefficient matrix. As shown in Fig. 3.9, this model uses the reconstruction coefficient matrix $\mathbf{W}$ to reconstruct the left half faces with the information of the right half faces. Since the raw data of images always contains slight noise, the Frobenius norm is utilized to deal with the noise in the data. Then the problem (3.7) can be rewritten as:

$$\min_{\mathbf{W},\mathbf{E}} ||\mathbf{E}||_F^2 + \alpha||\mathbf{W}||_p, \quad \text{ s.t. } \mathbf{X} = \widetilde{\mathbf{X}}\mathbf{W} + \mathbf{E}, \tag{3.8}$$

by removing the reconstruction error $\mathbf{E}$, the problem is equivalent to:

$$\min_{\mathbf{W}} ||\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}||_F^2 + \alpha||\mathbf{W}||_p, \tag{3.9}$$

where the first term represents the reconstruction residual and the second term is the constraint to the coefficient matrix. The values of two terms are balanced by the parameter α.

For the constraint of $\mathbf{W}$, each feature is supposed to be reconstructed with a few most representative features. Therefore, $l_{2,1}$-norm minimization is applied to keep the coefficient matrix sparse in rows. The $l_{2,1}$-norm of $\mathbf{W}$ can be replaced by the sum of 2-norm of all row vectors. Let $\mathbf{w}_i$ represent a row vector of $\mathbf{W}$. The

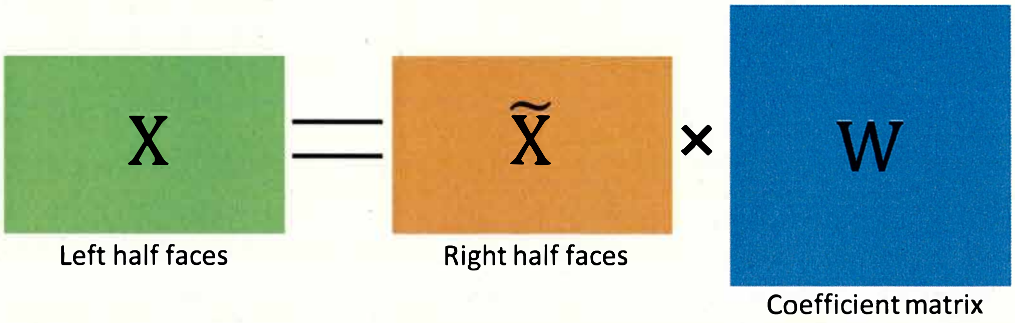


Figure 3.9: The basic model uses the coefficient matrix $\mathbf{W}$ to reconstruct the left half faces with the information in the right half faces.

corresponding reconstruction problem can be rewritten as:

$$\min_{\mathbf{W}} \|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2 + \alpha \sum_{i=1}^{d} \|\mathbf{w}_i\|_2 . \tag{3.10}$$

3.3.2 Optimization

The coefficient matrix $\mathbf{W}$ is the variable needs to be solved. The minimization problem Eq. 3.10 is equivalent to computing the minimum of the following function of $\mathbf{W}$:

$$\mathcal{L}(\mathbf{W}) = \|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2 + \alpha \sum_{i=1}^{d} \|\mathbf{w}_i\|_2 , \tag{3.11}$$

where both the first component and the second component are relevant to $\mathbf{W}$. To minimize the value of $\mathcal{L}(\mathbf{W})$, the derivative of $\mathcal{L}$ respect to $\mathbf{W}$ will set to be zero to construct the equation. Firstly, the derivative of $\mathcal{L}(\mathbf{W})$ is equivalent to the following form:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial\left(\|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2 + \alpha \sum_{i=1}^{d} \|\mathbf{w}_i\|_2\right)}{\partial \mathbf{W}} . \tag{3.12}$$

To simplify the above derivative, the relationship between the Frobenius norm and the trace of matrix can be used here, which can be written as follows:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\operatorname{Tr}(\mathbf{A}^*\mathbf{A})}, \tag{3.13}$$

where $\mathbf{A}$ is an arbitrary matrix with m rows and n columns, a_{ij} denotes its element in the i-th row and j-th column, and $\mathbf{A}^*$ denotes the conjugate transpose of $\mathbf{A}$. When the elements of A are all real numbers, the conjugate transpose of $\mathbf{A}$ is also equivalent to the transpose of $\mathbf{A}$. Hence, the Frobenius norm in the first component of $\mathcal{L}(\mathbf{W})$ is equivalent to the following form:

$$\begin{aligned}\|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2 &= \operatorname{Tr}\left(\left(\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\right)^T\left(\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\right)\right)\\ &= \operatorname{Tr}\left(\left(\mathbf{X}^T - \left(\widetilde{\mathbf{X}}\mathbf{W}\right)^T\right)\left(\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\right)\right).\end{aligned} \tag{3.14}$$

The transpose of the product of two matrices can be written as follows:

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T. \tag{3.15}$$

Hence, the first component of $\mathcal{L}(\mathbf{W})$ is equivalent to the following form:

$$\begin{aligned}\|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2 &= \operatorname{Tr}\left(\left(\mathbf{X}^T - \mathbf{W}^T\widetilde{\mathbf{X}}^T\right)\left(\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\right)\right)\\ &= \operatorname{Tr}\left(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\widetilde{\mathbf{X}}\mathbf{W} - \mathbf{W}^T\widetilde{\mathbf{X}}^T\mathbf{X} + \mathbf{W}^T\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\mathbf{W}\right).\end{aligned} \tag{3.16}$$

To simplify the above formula, the derivative of the product of two matrices can be written as follow:

$$\frac{\partial \operatorname{Tr}\left(\mathbf{B}^T\mathbf{A}\right)}{\partial \mathbf{B}} = \mathbf{A}, \tag{3.17}$$

and the derivative of the product of three matrices can be written as follow:

$$\frac{\partial \operatorname{Tr}\left(\mathbf{B}^T\mathbf{A}\mathbf{B}\right)}{\partial \mathbf{B}} = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{B}. \tag{3.18}$$

Then, the derivative of the first component of $\mathcal{L}\left(\mathbf{W}\right)$ is equivalent to the following form:

$$\frac{\partial \|\mathbf{X} - \widetilde{\mathbf{X}}\mathbf{W}\|_F^2}{\partial W} = -2\widetilde{\mathbf{X}}^T\mathbf{X} + 2\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\mathbf{W}. \tag{3.19}$$

The derivative of $\mathcal{L}\left(\mathbf{W}\right)$ is equivalent to the following form:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = 2\left(\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\mathbf{W} + \alpha\mathbf{P}\mathbf{W} - \widetilde{\mathbf{X}}^T\mathbf{X}\right), \tag{3.20}$$

where $\mathbf{P} \in \mathbb{R}^{d\times d}$ is a diagonal matrix, with its element $\mathbf{P}_{ii}$ calculated by:

$$\mathbf{P}_{ii} = \frac{1}{2\sqrt{\mathbf{w}_i^T\mathbf{w}_i + \delta}}. \tag{3.21}$$

Let the derivative of $\mathcal{L}$ respect to W to be zero, then the equation will constructed as follows:

$$2\left(\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\mathbf{W} + \alpha\mathbf{P}\mathbf{W} - \widetilde{\mathbf{X}}^T\mathbf{X}\right) = 0. \tag{3.22}$$

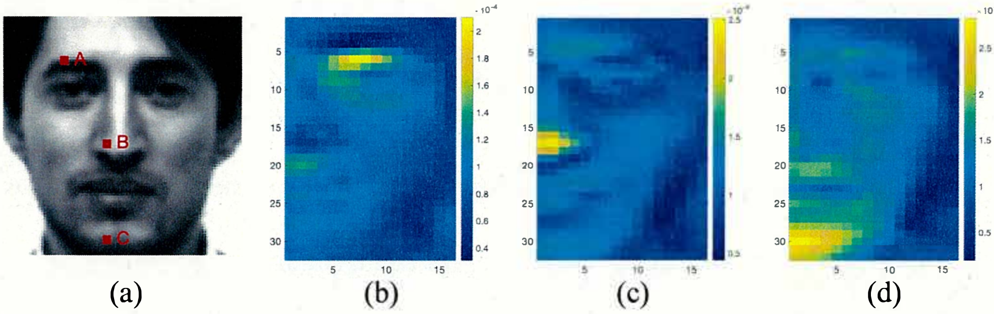
Considering that the coefficient matrix is sparse in rows, the term of $\mathbf{w}_i^T\mathbf{w}_i$ will probably obtain zero in theory. Therefore, we add a small enough value δ to avoid this situation. To obtain the solution of $\mathbf{W}$, the Eq. 3.22 can be rewritten as:

$$\mathbf{W} = \left(\widetilde{\mathbf{X}}^T\mathbf{X} + \alpha\mathbf{P}\right)^{-1}\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} \tag{3.23}$$

In this equation, variable $\mathbf{W}$ cannot be directly solved considering $\mathbf{P}$ also depends on the value of $\mathbf{W}$. Thus we propose an alternative iterative algorithm to find the optimal solution. When $\mathbf{W}$ is fixed, we can obtain $\mathbf{P}$ by Eq. 3.21 . When $\mathbf{P}$ is fixed, $\mathbf{W}$ can be easily obtained by Eq. 3.23. The procedure of our optimization algorithm is given by Algorithm 1.

Algorithm 1 Alternative iterative algorithm to compute the reconstruction coefficient matrix

```
Input: The data matrix for left-half face images $\mathbf{X} \in \mathbb{R}^{n \times d}$, the data matrix for
    right-half face images $\widetilde{\mathbf{X}} \in \mathbb{R}^{n \times d}$, the parameter $\alpha$, a small enough constraint
    $\delta$.
Output: The coefficient matrix $\mathbf{W}$.
Initialize $\mathbf{P} = \mathbf{I}, \mathbf{P} \in \mathbb{R}^{d \times d}$
Repeat
    Update $\mathbf{W} = \left(\widetilde{\mathbf{X}}^T \mathbf{X} + \alpha \mathbf{P}\right)^{-1} \widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}}$
    Update the diagonal matrix $\mathbf{P} \in \mathbb{R}^{d \times d}$ by
        $$\mathbf{P}_{ii} = \frac{1}{2\sqrt{\mathbf{w}_i^T \mathbf{w}_i + \delta}}, (i = 1, 2, \cdots, d).$$
until Convergence.
```



(a) (b) (c) (d)

Figure 3.10: Illustration of points on the left-half face being reconstructed with weighted features of the right-half face. (a) is the original facial image; (b) is the reconstruction weight matrix for point A; (c) is the reconstruction weight matrix for point B; (d) is the reconstruction weight matrix for point C.

3.3.3 Application

As shown in Fig. 3.10(a), if the facial information of the highlighted three points on the left-half face are corrupted, the information of these points A, B, C can be reconstructed with all the features of the right-half face. The reconstruction coefficient matrices calculated by Algorithm 1 are illustrated in heat maps. Take point A for example, the pixel value of point A can be estimated by a linear combination of all the pixels on the right-half face with the coefficients illustrated in Fig. 3.10(b). From these calculated coefficient matrices, it can be concluded that points close to the symmetrical position are given the largest weights, which indicates that these points are the most important ones to reconstruct points on the left face. If we only use the most important points to reconstruct points on the left face and ignore other less important points, the reconstruction process equals to mirroring/flipping a half face to the other half.

As an application of this characteristic, if there are occlusions in some areas of the left-half face, we can estimate the original information in these occluded areas using the reconstruction strategy described aforementioned. However, it is always unknown that either the left-half or the right-half is occluded. To address this problem, as shown in Fig.3.11, three cases have to be taken into account. First, the left half face is partially occluded. The features on the right face can be used to reconstruct the occlusion area. Second, the right half face is partially occluded. The features on the left face can be used to reconstruct the occlusion. Third, there is no occlusion on the face image and the reconstruction is not necessary.

Since the reconstruction for the first two cases needs all the facial images in a strictly cropped image dataset to calculate the coefficient matrix, the calculation is time-consuming and the reconstruction result also takes features with small weights into account which is likely to be sensitive to the noise on the right-half face image. According to the $l_{2,1}$-norm minimization in Eq. 3.10, the reconstruction coefficient matrix is constrained to be sparse in rows. This constraint tries to reconstruct a point on the left face with a very small number of points on the right face. To simplify the calculation, we propose to use the symmetrical point alone on the right-half face to reconstruct the corresponding point on the left face. This reconstruction strategy satisfies both the row sparsity of coefficient matrix and the point with maximum weight being selected according to results illustrated in Fig. 3.10. For this process, it is necessary to determine the midline of the face. Our face symmetry detection method used the Log-Gabor filter to extract the texture-orientation feature and the HSV color space to extract the color feature [25].

Therefore, the occluded face image can be reconstructed by coping the unoccluded half face and flipping it to the other side to replace the other half face, as shown in Fig.3.11 (b) and (c). Given an arbitrary frontal view face image I with width a, let $I(x, y)$ denotes the pixel with coordinates (x, y). In one case, flipping the left face generates a new image I_L using Eq. 3.3. In this case, the left-half face is selected as the candidate's unoccluded area. Then, flip the left-half face to the right side, and combined it with the original candidate area to construct a conjecture whole face.

In another case, the right-half face is selected as the candidate unoccluded region. Then, flip the right-half face to the left, and combined with the original candidate region to construct a conjecture whole face I_R using Eq. 3.4. Figure 3.11

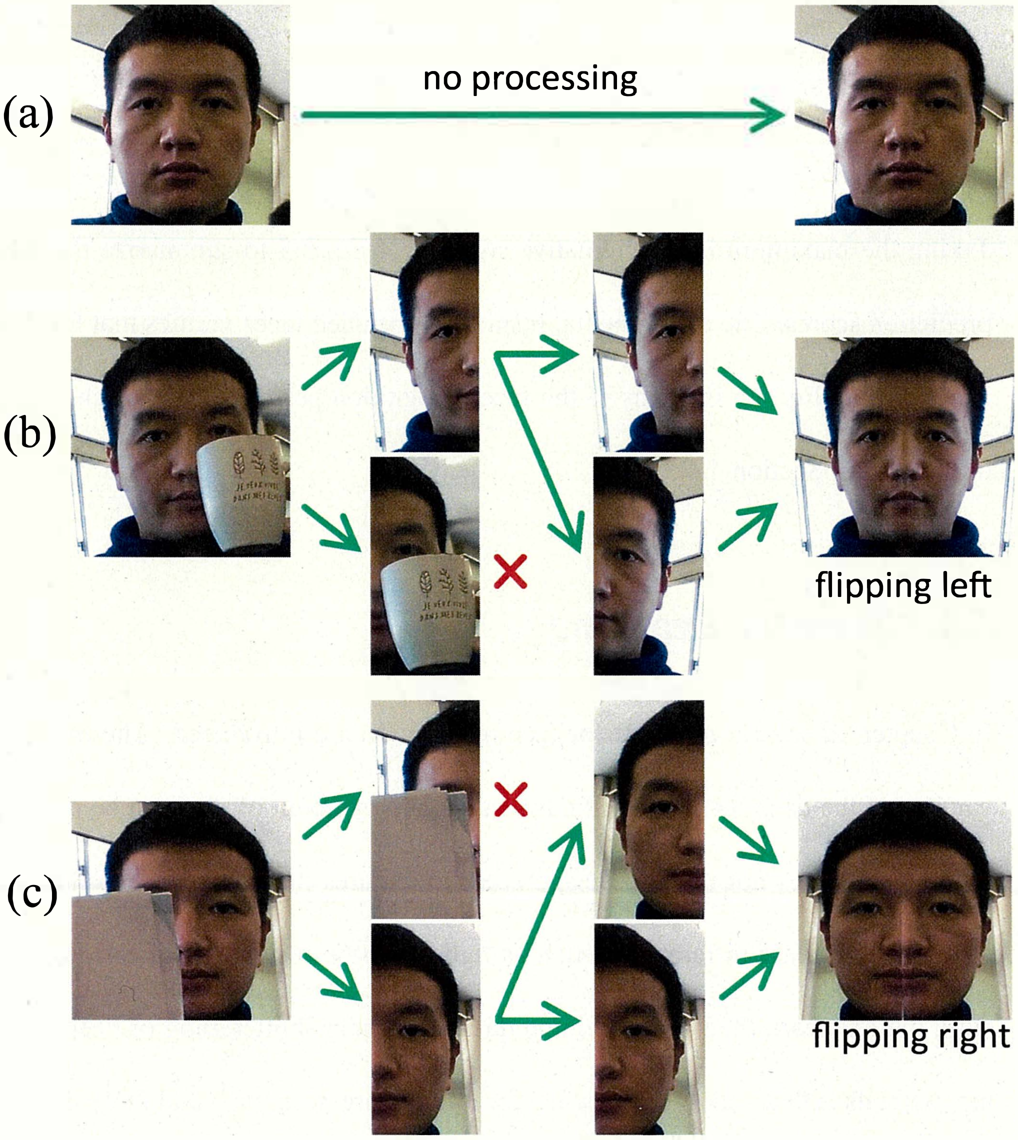


Figure 3.11: Three cases of reconstructing face images to discard the influence of partial occlusion. (a) is the face image that does not need any processing steps; (b) is the face image that right-half face occluded; (c) is the face image that left-half face occluded.

illustrate the step of flipping and the construction of two artificial faces. Finally, the three images are predicted by the classifier individually and the final prediction score $\hat{p}(I)$ takes the maximum:

$$\hat{p}\left(I\right) \leftarrow \max\left(p\left(I\right), p\left(I_L\right), p\left(I_R\right)\right). \tag{3.24}$$

Taking the maximum is an alternative strategy of Eq. 3.5 to summarize the three prediction scores. The experimental results on occluded faces verifies that Eq. 3.24 is better than Eq. 3.5 in terms of the face recognition performance, which will be introduced in Section 4.4.3.

3.4 Transfer Learning

In Chapter. 2, several methods for face recognition are introduced. The main advantage of the deep learning method is that a large amount of data can be used for training and does not need to design specific features that are robust to different types of intra-class differences (such as lighting, posture, facial expressions, age, etc.), but can learn them from training data. The main shortcoming of deep learning methods is that very large datasets for training are necessary, and these datasets need to contain enough changes so that they can be generalized to unseen samples. In our case, there are not enough uneven illumination and uneven occlusion data for training, so we utilize transfer learning method to do face recognition task. Transfer learning is defined as a method that extracts knowledge from one or more source tasks and applies the knowledge to a target task [59]. In our case, for a faster and easier experiment, instead of training a new deep network to obtain optimal weight parameters for each layer by illumination of a large face image dataset, we applied pre-trained neural network models that demonstrated good classification ability for the classification task on ImageNet. The parameter tuning process will be faster because the fundamental feature extraction layers have been trained. The pre-trained CNN models applied in this study include AlexNet, GoogLeNet, SqueezeNet, ResNet-50, Inception-v3, and DenseNet-201. Among them, AlexNet and GoogLeNet are presented as examples to explain the layer replacement and parameter configuration. AlexNet contains five convolutional layers with max-pooling used in the first, second, and fifth convolution layers, and three fully-connected layers. The important image features can be extracted using these layers. Relu is used as the activation function. In addition, local response normalization, dropout, and data augmentation have been performed in AlexNet, which has been used in many deeper neural networks. GoogLeNet has a deeper network structure and fewer parameters than AlexNet. Owing to the use of average pooling to replace the fully connected layer in the traditional network architecture and its well-designed inception architecture, GoogLeNet performs well in classification tasks and is widely used in transfer learning. All the pre-trained networks require the replacement of the final fully connected layer to fit the number of classes in the new task.

Figure 3.12 illustrates the process of transfer learning by AlexNet. The training process is illustrated in Fig. 3.13 We replaced the last three layers in AlexNet with our layers: a fully connected layer, softmax layer, and classification layer. The remaining parameters in the original model were preserved. Subsequently, the architecture was segmented into two: the pre-trained network and the trans-

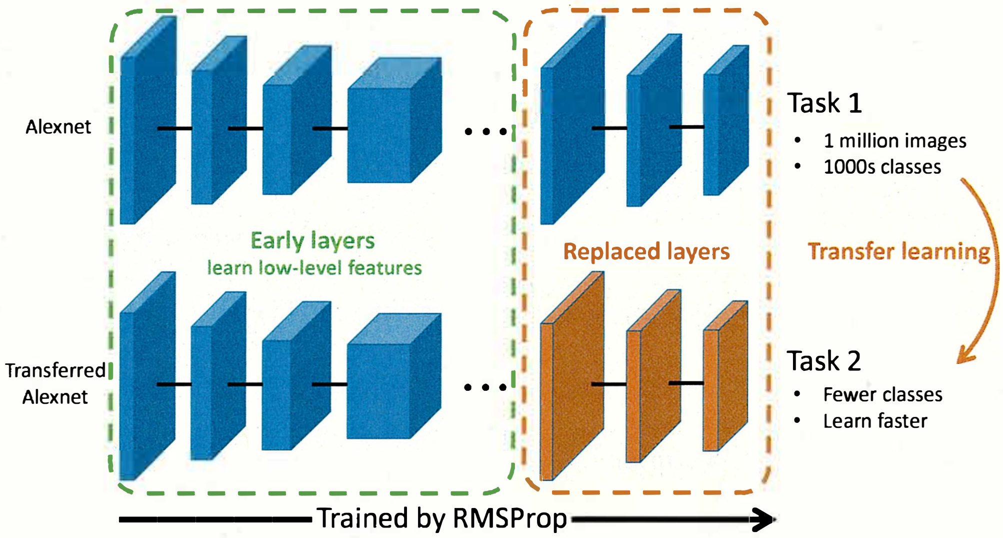


Figure 3.12: Transfer learning by AlexNet

ferred network. AlexNet has been trained on over a million images on ImageNet. Rich feature representations have been learned from a wide range of images. In the pre-trained AlexNet, the last three layers were configured for 1000 classes. For GoogLeNet, we replaced a fully connected layer and a classification layer. To transfer the pre-trained network to learn a new task, we specified the new options of the new layers according to the new data. As transfer learning employs all the parameters in the pre-trained network as initiation, it can exploit the features learned from massive images. Furthermore, the training of complex deep networks requires high-performance GPU and CPU, but the tasks using transfer learning can be implemented on ordinary personal computers. Therefore, the experiment is rendered more convenient and effective.

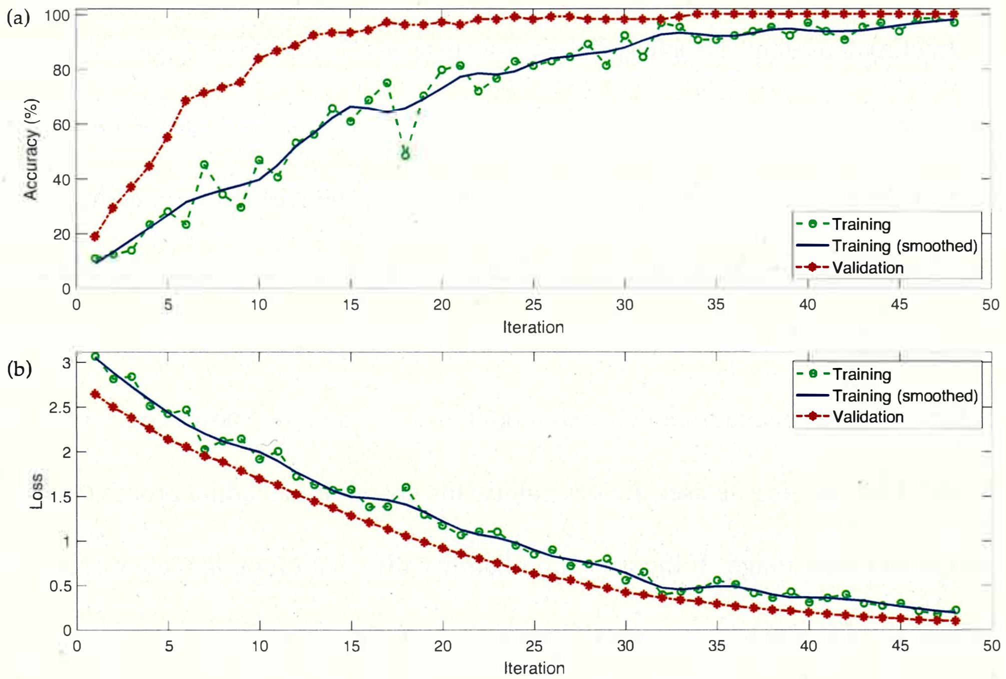


Figure 3.13: Illustration of training process. (a) shows the accuracy varies over the number of iteration; (b) shows the loss varies over the number of iteration.

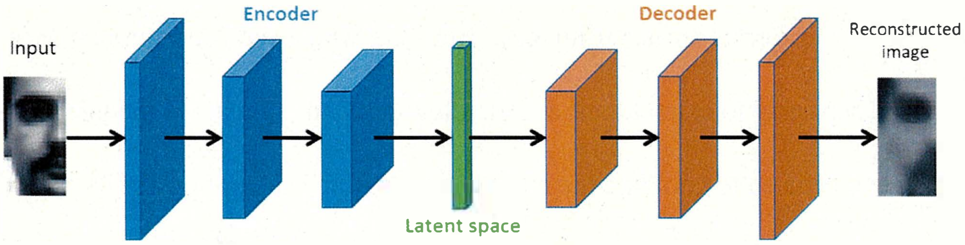


Figure 3.14: The framework of the variational autoencoder network for half face.

3.5 Facial Latent Spaces

A representative generative model of variational autoencoder [60] can be used for extracting the latent variables off the whole face and the half face. Figure 3.14 shows the frame work of the variational autoencoder network for half face [61]. If the latent variable of the whole face is similar to that of the half face, it will verify that the half face has similar separating ability to the whole face and the visualization of latent variables will illustrate the similarity between their separating ability. The variational autoencoder model is commonly used to generate new images in similar style of the training dataset. It does not use the encoding-decoding process to reconstruct an input image. It imposes a probability distribution on the latent space, and learns the distribution so that the distribution of outputs from the decoder matches that of the observed data. Then, the model will sample from this distribution to generate new data. The variational autoencoder model has two parts: the encoder and the decoder. The encoder takes an image input and outputs a compressed representation (the encoding), which is a vector of pre-defined length, which equals to 100 for the whole face images in this paper. The decoder takes the compressed representation, decodes it, and recreates the original image. In this thesis, the input data takes the format of facial images. Both the whole face and the half face are encoded separatly to extract their encoded low-dimentional data in the latent space, which can be used to compare the information loss and separating ability.

From the original face image size (e.g. $32 \times 32 \times 1$) to the pre-defined length of the latent space (e.g. 100), the dimension is substantially reduced. From this perspective, the variational autoencoder model can also be regarded as a dimensionality reduction technique, which selects a small number of features from the whole feature set, or create some new features based on the old ones. This process is found useful in some applications, which need low-dimensional data, such as data compression and data visualization. In the case of variational autoencoder, the encoder produces new features representation from the raw data, and the decoder performs as the reverse process. Original features in the raw data can be named with the initial space, and the new features which the encoder produces can be named with the encoded space, also called latent space. Depending on the initial space, the encoded space, and the reversed output of the decoder, the encoding process may lose a part of the information, which the decoding process cannot recover. In other words, the variational autoencoder model can be lossy while performing the dimensionality reduction. However, the variational autoencoder is designed to find the optimal pair of encoder and decoder from a given set of all possible encoders and decoders, and this optimal pair of encoder and decoder can remain as much information as possible. Thus, the reconstruction error between the decoded output and the initial features is supposed to be minimized, which can be written as follows:

$$(f_e^*, f_d^*) = \underset{(f_e, f_d) \in F_E \times F_D}{\arg\min} \; \epsilon\left(X, f_d\left(f_e\left(X\right)\right)\right), \tag{3.25}$$

where the X denotes the original features in the initial space, f_e, f_d represents the encoding function and the decoding function respectively, and the (f_e^*, f_d^*) represents the optimal encoding function and the decoding function pair. The reconstruction error between the decoded output and the initial features is defined by

$$\epsilon\left(X, f_d\left(f_e\left(X\right)\right)\right). \tag{3.26}$$

The autoencoder architecture is frequently used to solve the problem described in Eq. 3.25. The autoencoder sets both the encoding function and the decoding function as neural networks and learns the optimal pair of encoder and decoder with an iterative optimization process. As the optimization starts, the autoencoder architecture is fed with some training data, and the reconstruction error described in Eq. 3.26 is calculated and backpropagated through the architecture to update the parameters of the entire neural networks. When the iteration process tends to convergence, the optimal pair of encoder and decoder will be learned. The overall autoencoder architecture can be described as follows:

$$\begin{aligned}(f_e^*, f_d^*) &= \underset{(f_e,f_d)\in F_E\times F_D}{\arg\min} \quad ||\mathrm{X}-\hat{\mathrm{X}}||^2 \\ &= \underset{(f_e,f_d)\in F_E\times F_D}{\arg\min} \quad ||\mathrm{X}-f_d(\mathrm{Q})||^2 \\ &= \underset{(f_e,f_d)\in F_E\times F_D}{\arg\min} \quad ||\mathrm{X}-f_d\left(f_e\left(\mathrm{X}\right)\right)||^2,\end{aligned} \tag{3.27}$$

where Q denotes the encoded latent variables. The difference between the initial features and the reconstructed output of the autoencoder architecture is estimated by the l_2-norm.

However, the loss function described in Eq. 3.27 has a disadvantage that the model learns the pair of encoder and decoder to keep as much initial information as possible and ignores the interpretable and exploitable structures while training by the gradient descent strategy. The variational autoencoder model solves this issue by encoding the initial space as a distribution over the latent space. The variational autoencoder model can be trained as follows:

1. Firstly, assume the initial space can be encoded as a distribution $p\,(a \mid b)$ from a given distribution set, e.g. Gaussian distributions. Encode the raw data with the initial encoder networks to the latent space.

2. Secondly, sample a point from the distribution, which is also from the latent space.

3. Thirdly, decode the point and calculate the reconstruction error according to Eq. 3.27.

4. Fourthly, backpropagate the reconstruction error through the encoding networks and the decoding networks and update their weight parameters.

5. Finally, repeat the second step, the third step, and the fourth step until convergence.

In the first step, the assumed distribution is usually defined to be normal, which will make the encoding networks output the mean and the covariance matrix to describe the distribution set. To ensure the distribution to be close to a standard normal distribution, a regularization term, expressed as the Kulback-Leibler divergence, can be added to Eq. 3.27 and the overall model can be written as:

$$(f_e^*, f_d^*) = \underset{(f_e, f_d) \in F_E \times F_D}{\arg\min} \|\mathrm{X} - f_d\,(f_e\,(\mathrm{X}))\,\|^2 + D_{KL}\,(\mathcal{N}\,(\mu_x, \sigma_x)\,\|\mathcal{N}\,(0, 1)) \tag{3.28}$$

where $\mathcal{N}(\mu_x, \sigma_x)$ denotes a normal distribution with the mean of μ_x, and the standard deviation of σ_x. The Kulback-Leibler divergence is denoted by $D_{KL}(\cdot)$. The calculation of the Kulback-Leibler divergence while using distribution q to approximate distribution p can be expressed as follows:

$$\begin{aligned} D_{KL}(p\|q) &= \sum_{i=1}^{N} p(x_i) \cdot (\log p(x_i) - \log q(x_i)) \\ &= \sum_{i=1}^{N} p(x_i) \cdot \log \frac{p(x_i)}{q(x_i)}, \end{aligned} \tag{3.29}$$

which can also be expressed in the form of expectation:

$$D_{KL}(p\|q) = E\left[\log p(x) - \log q(x)\right]. \tag{3.30}$$

If the logarithm chooses the base of 2, then Kulback-Leibler divergence represents the information loss measured by binary digit while using q to approximate p.

To approximate complex distributions, the variational autoencoder applies the technique of variational inference, which finds the best approximation from an assumed distribution family. For instance, the latent space is firstly assumed to follow Gaussian distribution. The Gaussian distribution family can be expressed by the parameters of the mean and the covariance matrix. Then, the technique of variational inference can be used to find the best pair of the mean and the covariance matrix to approximate the target distribution in the latent space. To explain how it finds the best approximation, a probabilistic assumption has to be introduced. Assume the data variable $a \in \mathrm{X}$ is generated from a latent variable $b \in \mathrm{Q}$, then this assumption can be described as follows: the data variable a is sampled from the conditional likelihood distribution $p\left(a \mid b\right)$. The conditional likelihood distribution $p\left(a \mid b\right)$ describes the decoding process and $p\left(b \mid a\right)$ describes the encoding process. Applying the Bayes theorem, we have

$$p(b \mid a) = \frac{p(a \mid b)p(b)}{p(a)}. \tag{3.31}$$

Suppose $p(b \mid a)$ can be approximated by function $q_a\left(b\right)$ which follows a Gaussian distribution, the mean and the covariance of $q_a\left(b\right)$ is defined by two aforementioned functions, μ_x and σ_x. Thus, this relationship can be expressed by

$$q_x \sim \mathcal{N}\left(\mu_x, \sigma_x\right). \tag{3.32}$$

Then the problem of finding the optimal distribution to approximate the latent space is transformed into finding the optimal pair of function μ_x and function σ_x. Meanwhile, $\mathcal{N}\left(\mu_x, \sigma_x\right)$ is supposed to be as close to the standard normal distribution as possible, which can be written as follows:

$$\left(\mu_x^*, \sigma_x^*\right) = \underset{(\mu_x, \sigma_x) \in G \times H}{\arg\min} \; D_{KL}\left(q_a\left(b\right) \| p\left(b \mid a\right)\right). \tag{3.33}$$

According to Bayes theorem expressed in Eq. 3.31, the Kulback-Leibler divergence in Eq. 3.33 can be rewritten as follows:

$$D_{KL}\left(q_a\left(b\right) \| p\left(b \mid a\right)\right) = \mathbb{E}_{b \sim q_a}\left(\log q_a\left(b\right)\right) - \mathbb{E}_{b \sim q_a}\left(\log \frac{p(a \mid b)p\left(b\right)}{p\left(a\right)}\right). \tag{3.34}$$

By splitting the expectations, we have

$$
\begin{aligned}
&D_{KL}\left(q_a\left(b\right), p\left(b \mid a\right)\right) \\
&= \mathbb{E}_{b\sim q_a}\left(\log q_a(b)\right) - \mathbb{E}_{b\sim q_a}(\log p(b)) - \mathbb{E}_{b\sim q_a}(\log p(a \mid b)) + \mathbb{E}_{b\sim q_a}(\log p(a)) \\
&= \mathbb{E}_{b\sim q_a}\left(\log q_a(b)\right) - \mathbb{E}_{b\sim q_a}(\log p(b)) - \mathbb{E}_{b\sim q_a}(\log p(a \mid b)) + e \\
&= D_{KL}\left(q_a(b) \| p(b)\right) - \mathbb{E}_{b\sim q_a}(\log p(a \mid b)) + e \\
&= D_{KL}\left(q_a(b) \| p(b)\right) - \mathbb{E}_{b\sim q_a}\left(-\frac{\|a - \zeta(b)\|^2}{2c}\right) + e,
\end{aligned}
\tag{3.35}
$$

where e is a constant. In the second line of 3.35, $p(a)$ is irrelevant with the distribution that b follows, thus $\mathbb{E}_{b\sim q_a}(\log p(a))$ can be represented as the constant e. In the third line, the Kulback-Leibler divergence is extracted according to Eq. 3.29. In the last line, the distribution of $p(a \mid b)$ is parameterized as follows:

$$
p(a \mid b) \equiv \mathcal{N}(\zeta(b), cI) \quad \zeta \in F \quad c > 0,
\tag{3.36}
$$

where ζ denotes the function which is used to approximate the mean, and c is the constant which is used to approximate the covariance. Therefore, combining Eq. 3.33 with Eq. 3.35, the model to find the optimal pair of function μ_x and function σ_x described in Eq. 3.33 can also be rewritten as:

$$
\begin{aligned}
(\mu_x^*, \sigma_x^*) &= \underset{(\mu_x, \sigma_x) \in G \times H}{\arg\min} \left(D_{KL}\left(q_a(b) \| p(b)\right) - \mathbb{E}_{b\sim q_a}\left(-\frac{\|a - \zeta(b)\|^2}{2c}\right)\right) \\
&= \underset{(\mu_x, \sigma_x) \in G \times H}{\arg\max} \left(\mathbb{E}_{b\sim q_a}\left(-\frac{\|a - \zeta(b)\|^2}{2c}\right) - D_{KL}\left(q_a(b) \| p(b)\right)\right),
\end{aligned}
\tag{3.37}
$$

where ζ is also a unknown function to be found. However, ζ can be optimized by maximizing the first term of the second line in Eq. 3.37. The overall variational autoencoder architecture involving the aforementioned three functions can be expressed in the following optimization model:

$$(\mu_x^*, \sigma_x^*, \zeta^*) = \underset{(\mu_x, \sigma_x, \zeta) \in G \times H \times J}{\arg\max} \left(\mathbb{E}_{b \sim q_a} \left(-\frac{\|a - \zeta(b)\|^2}{2c} \right) - D_{KL} \left(q_a(b) \| p(b) \right) \right), \tag{3.38}$$

where J is the function set from which the optimal ζ^* can be found. To this end, all the unknown variable can be solved by an optimization problem. Two deep networks can be used as the encoder network and the decoder network to solve this optimization problem. The experimental result of the optimization will be introduced in Section 4.5.

Chapter 4

Experiment

4.1 Datasets

4.1.1 Dataset for Experiments on Uneven Illumination

To validate the performance of the flipping scheme as a post-processing method, 660 images were augmented from the Yale dataset [62]. This dataset originally contains 165 images with 11 images for each individual. Each image was augmented to create four images by additional mirroring operations, simulating dark illumination for the original and mirrored images, respectively. Because each individual had two original images, the left–right image was unevenly illuminated, as shown in Fig. 1.1. The total number of left–right unevenly illuminated images was 120 after augmentation, which were selected for testing. The remaining 540 images were used for training, within which 162 (30%) images were used for validation and 378 (70%) images were used for training. The small frontal face dataset was used for this experiment as the focus of this study was on the development of a face recognition system for research laboratory members or family.

To validate the performance of the training strategy on half-face images, the Yale and CASIA-WebFace datasets [63] were used to test the flipping strategy as a preprocessing method. Because the face images of the CASIA-WebFace dataset were captured in the wild, only faces that can be detected by the OpenCV cascade frontal face detector were selected and cropped following the same standard of the Yale dataset to create the whole-face image dataset, which had 9,265 images. Compared with training the whole-face images, another 9,265 half-face images were cropped out to create the half-face image dataset. Both the whole-face and half-face datasets were segmented into 70% for training and 30% for testing. Table 4.1 shows the evaluation results of these two datasets for comparison.

4.1.2 Dataset for Experiments on Occlusion

For the occlusion problem, We used two datasets, the Yale facial dataset [62] and the extended Yale B facial dataset [64]. Yale facial dataset contains 165 gray scale images of 15 subjects. Each subject contains 11 images [62]. The extended Yale B facial dataset contains 16128 images of 28 human subjects. We used all of the images in the datasets for training the network. The test images was random selected from each individual.

4.2 Flipping Scheme for Testing

As the first step of our work, we loaded the pretrained AlexNet model and GoogLeNet. Subsequently, we examined the input dimension of the first layer of the CNN models and preprocessed the input images according to the model. In our study, AlexNet and GoogLeNet required input dimensions of $227 \times 227 \times 3$ and $224 \times 224 \times 3$, respectively. The primitive features from the images, such as edges and blobs, were learned by the first convolutional layer. Higher-level image features were formed by the deeper layers of the network combined with these low-level features.

As mentioned in Section 4.1.1, we used 15 individuals. Every individual had 11 images. Because we focused on face recognition under the effect of illumination, we selected two images for each individual, i.e., the left-lighter face image and right-lighter face image, as the final recognition images. The remaining nine non-illuminated images were used as training images. Because each individual had few images for training, we used three methods, i.e., mirroring, color equalization, and slight rotation to obtain more training data. Nine images were expanded using each method. For each individual, we expanded 27 images. In other words, we trained our own model using 36 images for each individual.

In this experiment, both AlexNet and GoogLeNet applied stochastic gradient descent with momentum with a mini-batch size of 20, an L2 regulation factor of 1×10^{-4}, and a momentum of 0.95. The maximum number of epoch was set to 6. The initial learning rate was 1×10^{-4}. Validation was performed for every epoch.

The pretrained CNN models can easily extract the training feature. We used cross-validation to randomly select images for repeated training and testing. This method is effective in testing the performance of our model and prevents overfitting. We can guarantee that our model performs well on the dataset during training by dividing the dataset into a train set and a validation set. Because this experiment was

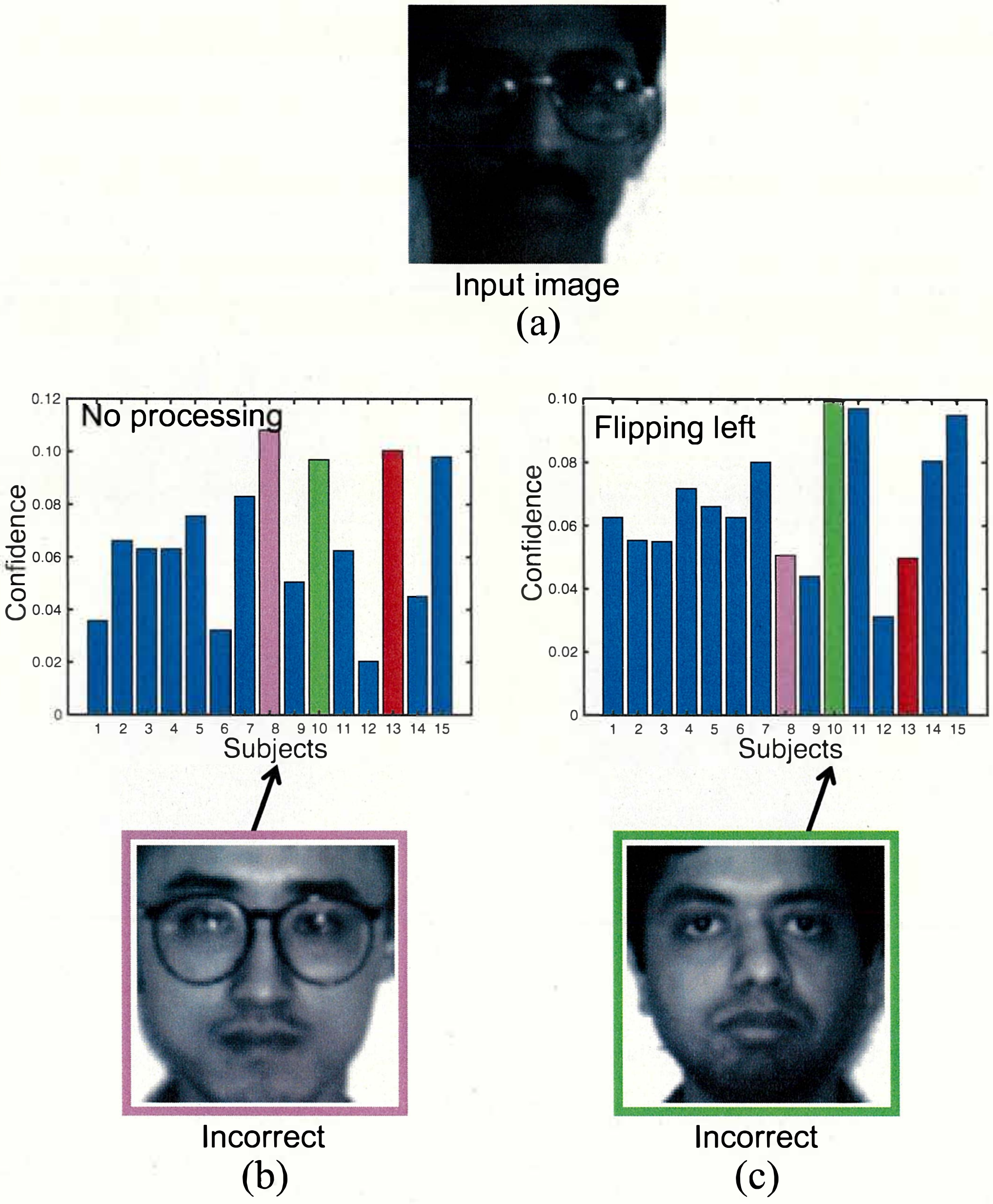


Figure 4.1: An example of classification result predicted by the first two methods. (a) Input image; (b) individual of 8, which is the result of the original classifier recognition of AlexNet; (c) individual of 10, which is the result of recognition after flipping left. Confidence refers to the prediction score of a test image belonging to a certain identity.

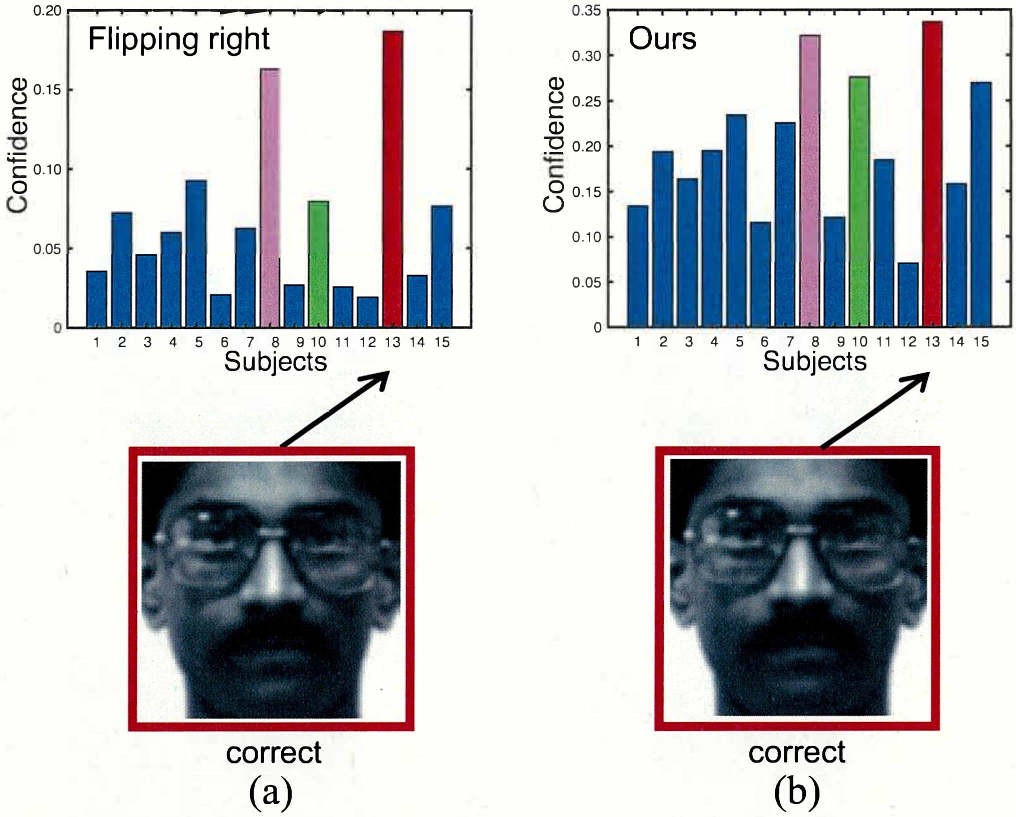


Figure 4.2: An example of classification result predicted by the other two methods. (a) individual of 13, which is the result after flipping right; (b) individual of 13, which is the result that sums the no-processing, flipping left, and flipping right. Confidence refers to the prediction score of a test image belonging to a certain identity.

designed to evaluate the flipping scheme as a post-processing method in the testing stage and was not related to the training process, three different classifiers for both AlexNet and GoogLeNet were obtained according to different maximum epoch settings, i.e., 4, 5, and 6. All classifiers were used to predict the artificial face image and the original test image for comparison. As illustrated in Figs. 4.3, Figs. 4.4, and Figs. 4.5, the receiver operating characteristic (ROC) curves were used to evaluate the flipping scheme. In these ROC curves, the curve of "no processing" represents the recognition accuracy obtained by inputting the original image into the AlexNet without any processing, and the curve of "flipping left" refers to the score that uses the image obtained by flipping the left-half face to the right and covering the original right side as the test image. The curve of "flipping right" refers to the score that uses the image obtained by flipping the right half face to the left and covering the original left side as the test image. The curve of "proposed method" refers to the score that sums the scores of "no-processing", "flipping left", and "flipping right". The three experiments shown in Figs. 4.3, Figs. 4.4, and Figs. 4.5 was performed on the classifier obtained on the AlexNet model while the maximum epoch is set as 4, 5, and 6 respectively. Similarly, in Figs. 4.6, Figs. 4.7, and Figs. 4.8, the curve of "no-processing" represents the recognition accuracy obtained by inputting the original image into the GoogLeNet model without any processing. the curve of "flipping left" and the curve of "flipping right" as well as the curve of "the proposed method" have the similar meaning as curves with AlexNet. Besides, the three experiments shown in Figs. 4.6, Figs. 4.7, and Figs. 4.8 was performed on the classifier obtained on the GoogLeNet model while the maximum epoch is set as 4, 5, and 6 respectively.

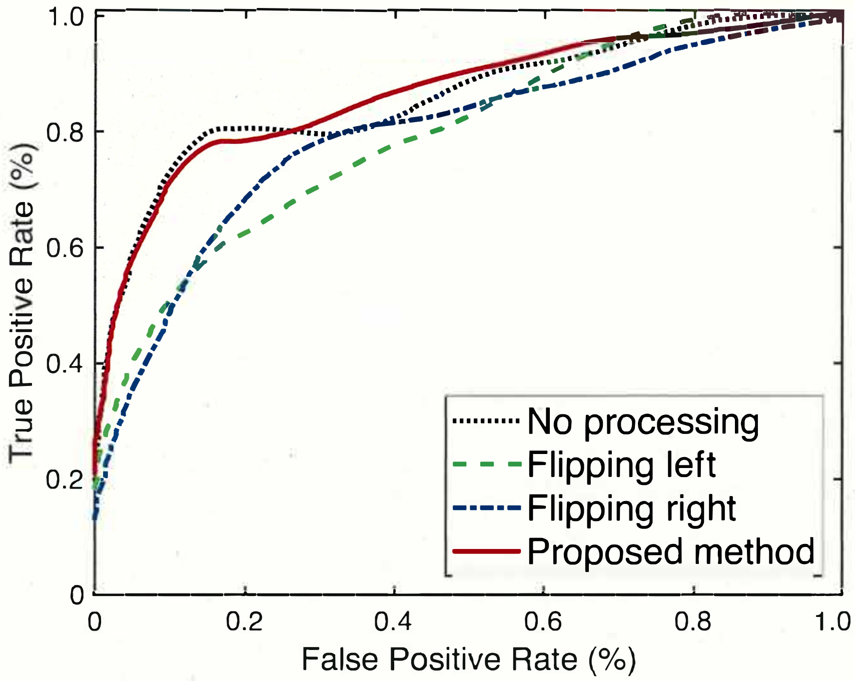


Figure 4.3: Performance comparison of four methods on three different classifiers transferred from **AlexNet** model with the maximum epoch of 4.

As shown in Figs. 4.3, Figs. 4.4, and Figs. 4.5, although the method of using flipping left and flipping right alone for recognition did not yield the best performance, the proposed method, which summarizes the prediction score of the other three methods for a final decision, achieved a better performance than normal transfer learning without post-processing. In Figs. 4.6, Figs. 4.7, and Figs. 4.8, the proposed method also achieved a better performance than normal transfer learning without post-processing.

Hence, we conclude that our proposed method can achieve the maximum accuracy in comparison with existing CNN models without post-processing. Our proposed method offers three main advantages. First, this method is simple and straightforward. It can be used as a post-processing step to the classifier trained by

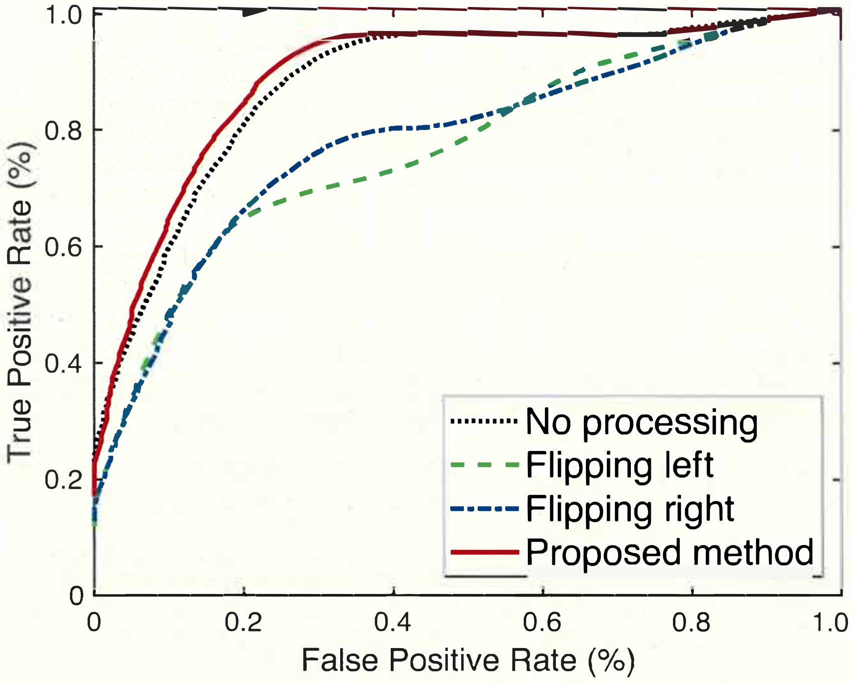


Figure 4.4: Performance comparison of four methods on three different classifiers transferred from **AlexNet** model with the maximum epoch of 5.

transfer learning, with a training cost of 11.32 min for Experiment 1. The other advantage is that we need not consider whether the training dataset is sufficiently large. Even with few training images, our method can reduce the effect of left–right uneven illumination on face recognition. The last advantage is that we need not consider the side with a bad illumination condition. The result obtained by calculating the sum score of the no-processing, flipping left, and flipping right can result in the correct recognition.

To verify the case where the flipping scheme achieved a better performance than normal testing without post-processing, an example is shown in Fig. 4.1 and Fig. 4.2 where the left–right unevenly illuminated testing image is classified as an incorrect identity by AlexNet according to the prediction score distribution. Furthermore, us-

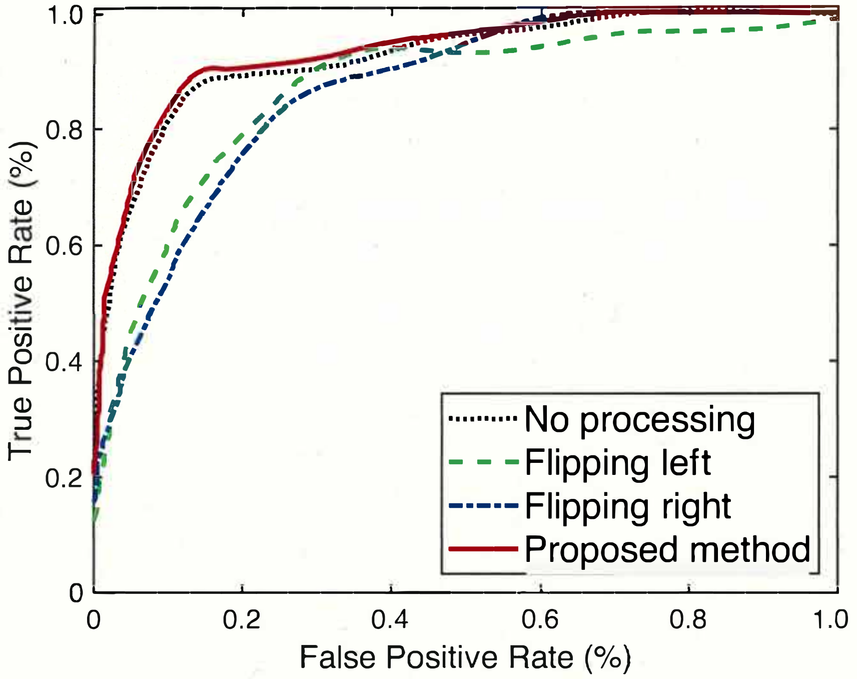


Figure 4.5: Performance comparison of four methods on three different classifiers transferred from **AlexNet** model with the maximum epoch of 6.

ing the flipping-left method that inputs the conjecture face flipping from left to right resulted in an incorrect identity. However, using the flipping-right method and the proposed method, which maximizes the aforementioned three score distributions, the correct identity was obtained, as shown in Fig. 4.2. It was clear that both the results of the no-processing and flipping-left method were wrong in this case. The flipping-right method yielded the correct recognition result, which rendered the sum score of our proposed method the correct one.

In this example, the flipping-left method caused both the left and right sides to be dark, which affected the performance of the CNN model. However, the clear right side was copied to the left side in the flipping-right method, rendering the appearance of the input image more similar to the correct identity. Therefore, we used

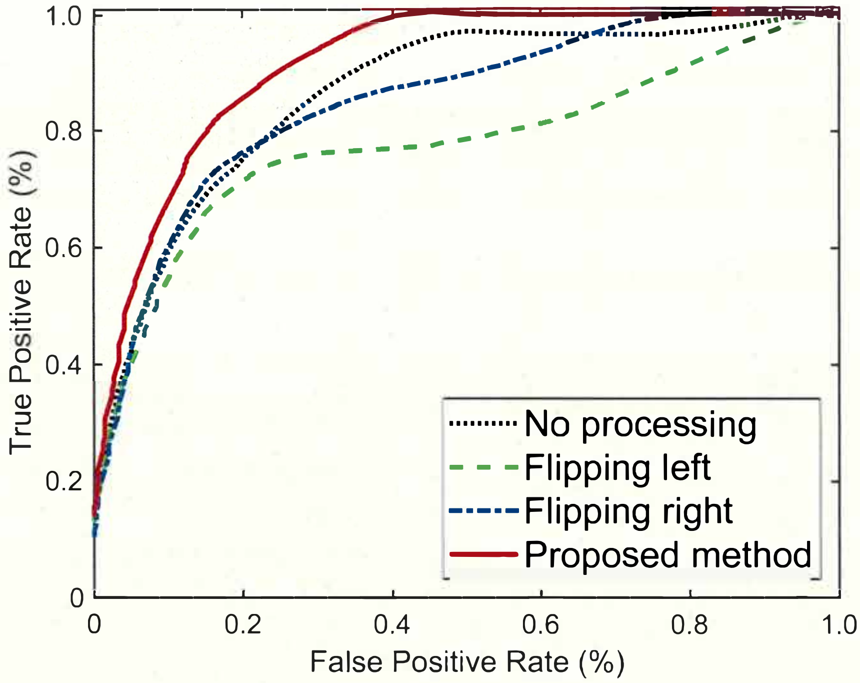


Figure 4.6: Performance comparison of four methods on three different classifiers transferred from **GoogLeNet** model with the maximum epoch of 4.

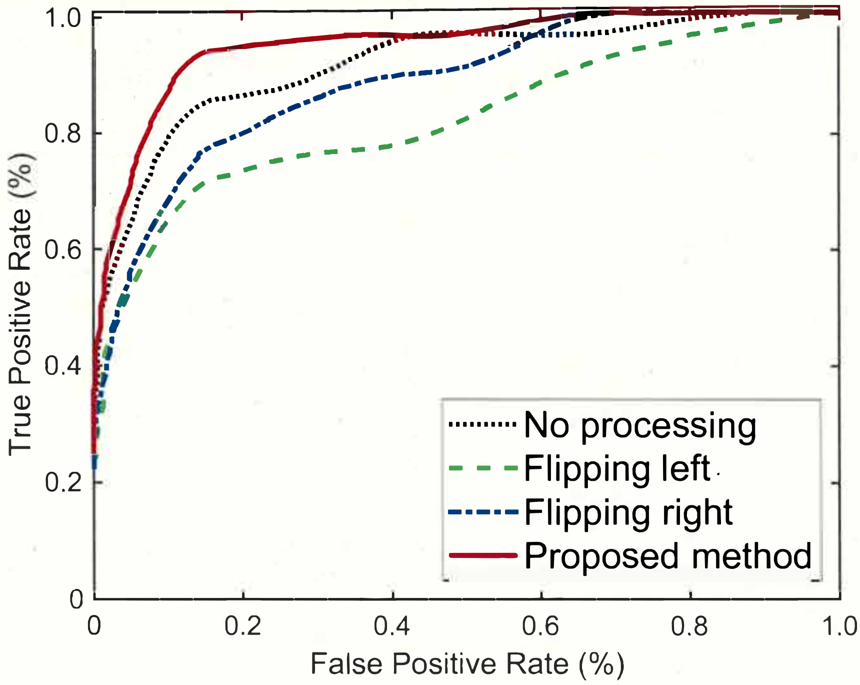


Figure 4.7: Performance comparison of four methods on three different classifiers transferred from **GoogLeNet** model with the maximum epoch of 5.

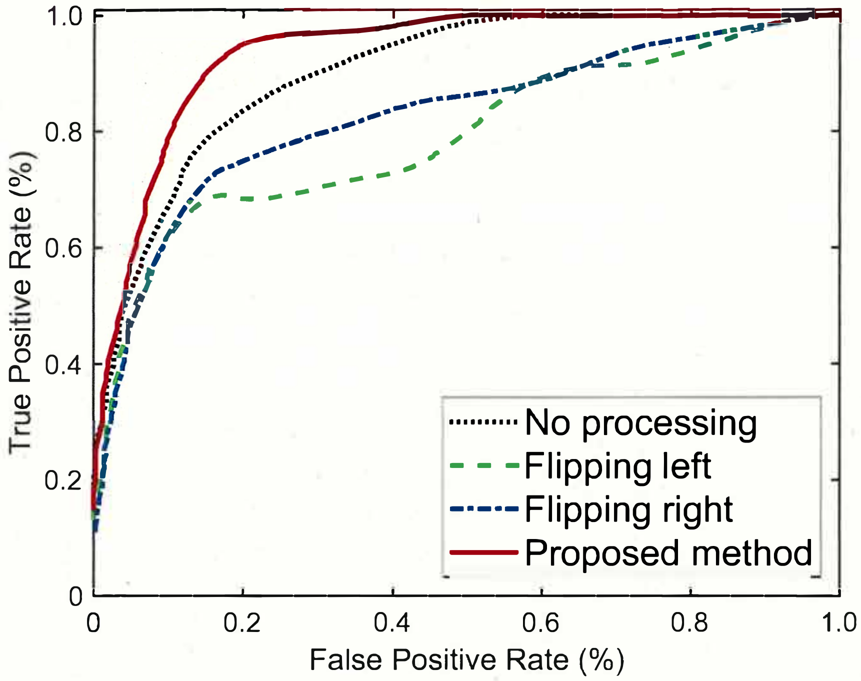


Figure 4.8: Performance comparison of four methods on three different classifiers transferred from **GoogLeNet** model with the maximum epoch of 6.

a flipping scheme to post-process the face image in our proposed method to improve the accuracy of face recognition in the case of left–right uneven illumination.

4.3 Training with Half Faces

The face reconstructed using the symmetrical characteristic of the face can effectively manage the left–right uneven illumination condition. When the face image is relatively regular and symmetrical, the face obtained by the flipping scheme is natural and appropriate for the recognition task. Meanwhile, other cases exist where the face image shows an exaggerated expression, rendering a nonsymmetrical face. The reconstructed faces in these cases may appear unnatural and different from a real face. Furthermore, another method is proposed herein to apply the symmetrical characteristic, i.e., training on half-face images. The CNN models learn from only half of the strictly cropped face image.

Training on both the whole-face and half-face images was performed for comparison. Both strategies use 9,265 images for training and testing, as described in Section 4.1.1. Six pretrained CNN architectures, i.e., AlexNet, GoogLeNet, SqueezeNet, ResNet-50, Inception-v3, and DenseNet-201 were used in transfer learning with both strategies to perform facial image classification on the two strictly cropped datasets. Each dataset contained 9265 images. The difference was that one dataset contained strictly cropped frontal view face images washed up from Yale and CASIA, whereas the other dataset contained half-faces cropped along the extracted symmetry, in which the right half-face images were flipped to obtain the same appearance as the left half-face.

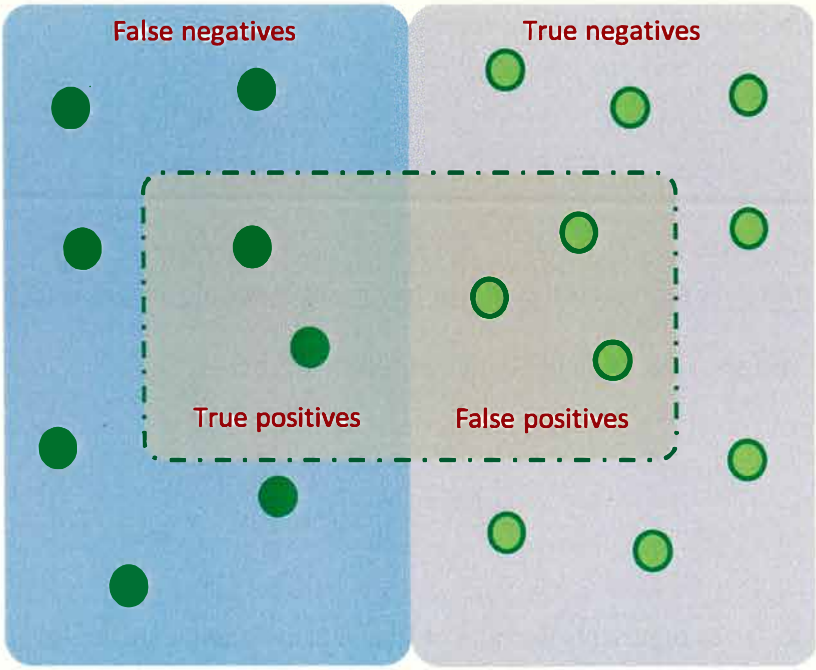


Figure 4.9: Four cases of predicted sample images.

The 12 ROC curves for six models with two training strategies were indistinguishable from each other. Evaluation metrics $accuracy$, $sensitivity$, $specificity$, and $AUCscore$ were used in this study for all the testing images (2779 images). Suppose the test sample images have been predicted into two categories of positives and negatives, and each category has samples which are predicted right and also has samples which are predicted wrong. Let TN, TP, FN, and FP denotes the true negatives, true positives, false negatives, and false positives, as shown in Fig 4.9. The true positives represent the sample images which are positives and predicted as positives. The false negatives represent the sample images which are positives and predicted as negatives. The true negatives represent the sample images which are negatives and predicted as negatives. The false positives represent the sample images which are negatives and predicted as positives. The $accuracy$ represents the ratio of sample images which are predicted right to all the sample images, which can be written as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}. \tag{4.1}$$

The $sensitivity$ represents the ratio of true positive sample images to all the positive sample images, which can be written as follows:

$$sensitivity = \frac{TP}{TP + FN}. \tag{4.2}$$

The $specificity$ represents the ratio of true negative sample images to all the negative sample images, which can be written as follows:

$$specificity = \frac{TN}{FP + TN}. \tag{4.3}$$

The $AUCscore$ represents the area under the ROC curve. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. An ROC curve plots true positive rate vs. false positive rate at different classification thresholds. The $AUCscore$ is an evaluation metric of the ROC curve. The larger value of $AUCscore$ represents the better classifier.

The PC used for the experiment had a CPU of i7-8700 (3.19 GHz), RAM of 16 GB, and GPU of GeForce GTX 1650. The results for these evaluation metrics are presented in Table 4.1. As shown in the table, all six pretrained CNN models were evaluated with similar input image sizes. The second column shows the number of layers in the source code level. Each model was evaluated using both the whole-face and half-face strategies. By comparing the results of two different strategies, it can be concluded that the half-face training strategy outperformed the whole-face training strategy on AlexNet, SqueezeNet, GoogLeNet, and Inception-v3 in terms of the $accuracy$, $sensitivity$, $specificity$, and $AUCscore$. The ResNet-50 and DenseNet-201 demonstrated a slightly worse performance on the half-faces than on the whole face.

Although training on the half-face images did not yield a better performance than training on the whole-face images using ResNet-50 and DenseNet-201, the other four models yielded better performances in terms of accuracy, sensitivity,

Table 4.1: Evaluation of transfer learning on **the whole face image dataset** vs. **the half face image dataset** with different pre-trained CNN models.

Model	#Layers	Input size	Strategy	Accuracy	Sensitivity	Specificity	AUC
AlexNet	25	227×227	whole face	0.962	0.699	0.980	0.965
			half face	0.986	0.886	0.992	0.996
SqueezeNet	68	227×227	whole face	0.977	0.819	0.988	0.986
			half face	0.984	0.874	0.992	0.994
GoogLeNet	144	224×224	whole face	0.984	0.876	0.992	0.996
			half face	0.990	0.889	0.995	0.997
ResNet-50	177	224×224	whole face	0.998	0.981	0.999	0.998
			half face	0.992	0.934	0.996	0.996
Inception-v3	316	299×299	whole face	0.990	0.922	0.995	0.998
			half face	0.997	0.976	0.998	0.999
DenseNet-201	709	224×224	whole face	0.999	0.991	0.999	0.999
			half face	0.995	0.961	0.997	0.997

specificity, and AUC. The accuracy of training on the whole face using ResNet-50 and DenseNet-201 exceeded 99%. Training on the half-face images yielded a slightly worse performance, i.e., less than 1%. Therefore, the classification performance by training on half-face images is comparable to that by training on the whole-face images.

4.4 Occlusions on Face

4.4.1 Occlusions on the Whole Face

Since fine-tuning a pretrained network is effective for extracting the knowledge from one source task and applies the knowledge to a target task [59], the pretrained series AlexNet was used for the classification tasks of faces with different degrees of occlusion in Sect. 4.4, Sect. 4.4.2 and Sect. 4.4.3. It consists of eight layers, five convolutional layers, and three fully connected layers. The last three layers were replaced by new layers: a fully connected layer, softmax layer and classification layer. The transferred network was trained by root mean square prop. The mini-batch size was set by 15 and the max epoch was 15. The momentum was 0.9 and the initial learning rate was 1×10^{-5}.

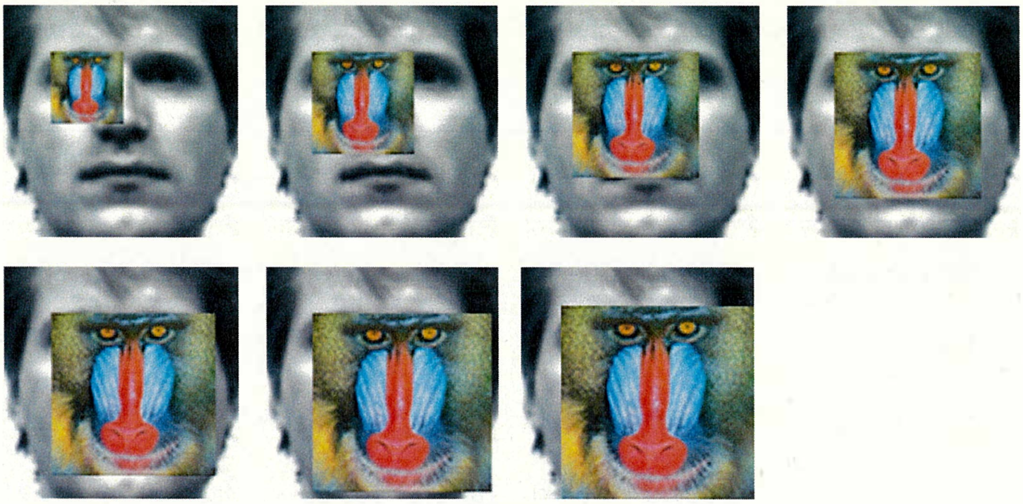


Figure 4.10: Examples of occlusions on the whole face.

In this experiment, two images of each individual were selected from the datasets to be processed as the test images. Sect. 4.4.2 and Sect. 4.4.3 were also the same as Sect. 4.4 by selecting two images from each individual to be processed as test images. In order to verify the effectiveness of the flipping scheme in different face occlusion ratios, we occluded each complete face from 10% to 70%. Here, we have seven groups of test images as shown in Fig.4.10. In the first part of this experiment, we evaluated the recognition rate of each group of occluded faces, as shown in Table 4.2 and Table 4.3. "No processing" represents the recognition accuracy obtained by inputting the original image into the classifier obtained by AlexNet without any processing, and "flipping left" refers to the score that uses the image obtained by flipping the left-half face to the right and covering the original right side as the test image. "Flipping right" refers to the score that use the image obtained by flipping the right-half face to the left and covering the original left side as the test image. "Proposed method" refers to the maximum score of the three cases as explained in Eq. 3.24.

Table 4.2: Performance comparison (%) on the extended Yale B database under varying levels of block occlusion.

Percent occluded	10%	20%	30%	40%	50%	60%	70%
No processing	100.00	100.00	100.00	89.47	68.42	47.37	38.16
Flipping left	64.47	44.74	43.42	35.53	19.74	17.11	11.84
Flipping right	100.00	100.00	100.00	100.00	97.37	84.21	71.05
Proposed method	100.00	100.00	100.00	100.00	82.89	69.74	59.21

From the results of Table 4.2 and Table 4.3 , the following conclusions can be drawn:

1. Occlusion seriously affects the recognition accuracy of AlexNet both on the two datasets. With the increase in occlusion ratio, the face recognition rate

Table 4.3: Performance comparison (%) on the Yale database under varying levels of block occlusion.

Percent occluded	10%	20%	30%	40%	50%	60%	70%
No processing	86.67	60.00	50.00	36.67	6.67	6.67	6.67
Flipping left	53.33	40.00	33.33	30.00	23.33	26.67	23.33
Flipping right	100.00	60.00	46.67	30.00	16.67	6.67	6.67
Proposed method	100.00	60.00	43.33	26.67	16.67	10.00	6.67

of AlexNet gradually decreases from 100% to 38.16% on the extend Yale B dataset and from 86.67% to 6.67% on the Yale dataset.

2. In all the occlusion image groups, “Proposed method” performs poorly in all the occlusion ratios both on the extended Yale B dataset. “Proposed method” performs well in all the occlusion ratio on the extended Yale B dataset. On the Yale dataset, “Proposed method” performed bad from the occlusion rate of 10% to 40%, but performed well from the occlusion rate of 50% to 70%.

3. On the extended Yale B dataset, “Proposed method” performed well in all the occlusion ratios. On the Yale dataset, the proposed method performs well when the face images are slightly occluded. However, when the occlusion ratio is high, the proposed method cannot outperform the result of “no processing”.

With the increase in occlusion ratio, more and more face feature will be corrupted. This is why the recognition rate of AlexNet without processing gradually decreases. As for the methods of flipping left and flipping right, when the non-occluded half face or less occluded face area is flipped to the other half and covered the original half, the recognition will be less affected by occlusion. The proposed method can significantly improve the performance of face recognition effected by partial occlusion problem.

4.4.2 Occlusion on the Half Face

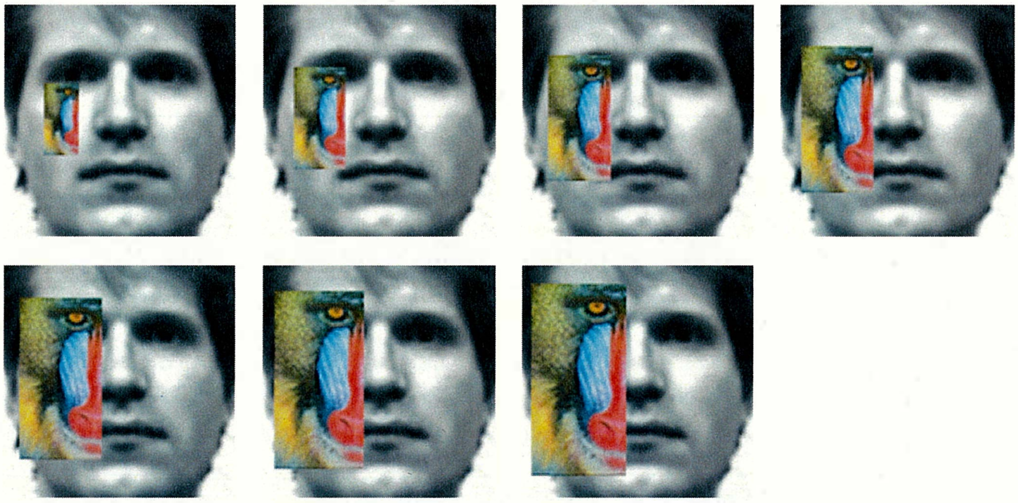


Figure 4.11: Examples of occlusions on the half faces.

In order to verify the recognition performance of flipping scheme when half of the face is occluded, we occluded each half face from 10% to 70%. Here, we have seven groups of test images as shown in Fig 4.11. In the first part of this experiment, we evaluated the recognition rate of each group of occluded faces, as shown in Table4.4 and Table4.5. From the results of the two table, the following conclusions can be drawn:

1. In all the occlusion image groups, "flipping left" performs poorly in all the occlusion ratios both on the two facial datasets. However, "flipping right" performs as well as 100% in each occlusion ratio on the Yale dataset.

2. On the extended Yale B dataset, “no processing” is less affected by occlusion and has a high recognition rate. “Proposed method” cannot outperform the AlexNet without processing in some occlusion ratios.

3. On the Yale dataset. “Proposed method” performs better than “no processing” and “flipping left”. However, “Proposed method” cannot outperform the flipping right.

Table 4.4: Performance comparison (%) on the extended Yale B database under varying levels of block occlusion on either half region (left-half or right-half).

Percent occluded	10%	20%	30%	40%	50%	60%	70%
No processing	100.00	100.00	100.00	97.37	97.37	92.11	90.79
Flipping left	72.37	63.16	55.26	47.37	34.21	25.00	18.42
Flipping right	88.16	88.16	88.16	88.16	88.16	88.16	88.16
Proposed method	100.00	100.00	98.68	97.37	96.05	93.42	88.16

Table 4.5: Performance comparison (%) on the Yale database under varying levels of block occlusion on either half region (left-half or right-half).

Percent occluded	10%	20%	30%	40%	50%	60%	70%
No processing	100.00	93.33	96.67	73.33	73.33	73.33	60.00
Flipping left	53.33	30.00	10.00	6.67	10.00	20.00	6.67
Flipping right	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Proposed method	100.00	100.00	100.00	96.67	100.00	100.00	100.00

The reason why the proposed method cannot outperform the flipping right is that this method considers both the flipping left case and flipping right case. Since the left-half face is occluded in our experiment, flipping right method eliminates the bad effects, so it performs better than flipping left method. On the other hand, in the case where half-face is partially occluded, the recognition result which got from the large dataset trained CNN model is less affected by the occlusion. However, the recognition model of the CNN trained by the smaller Yale dataset will have a greater effect by occlusion. In the case that half-face is partially occluded, our proposed method performs better on the small dataset.

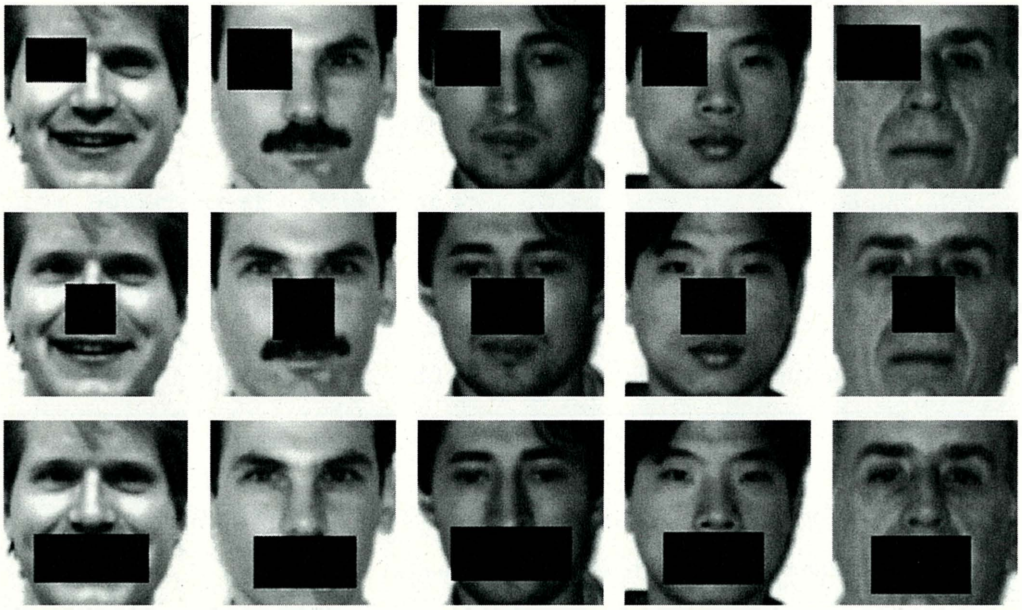


Figure 4.12: Occlusion on eye, nose and mouth.

4.4.3 Occlusion on the Eye, Nose and Mouth

In order to find out which part of human face is the most important factor during face recognition, we occluded the area of eye, nose, and mouth, respectively. Here, we have three groups of test images as shown in Fig. 4.12. In the first part of this

Table 4.6: Performance comparison (%) on the Yale database under occlusion on the area of eye, nose, and mouth.

Occluded area	eye	nose	mouth
No processing	46.67	53.33	40.00
Flipping left	20.00	40.00	33.33
Flipping right	80.00	33.33	40.00
Average score	66.67	40.00	40.00
Maximum score	66.67	46.67	40.00

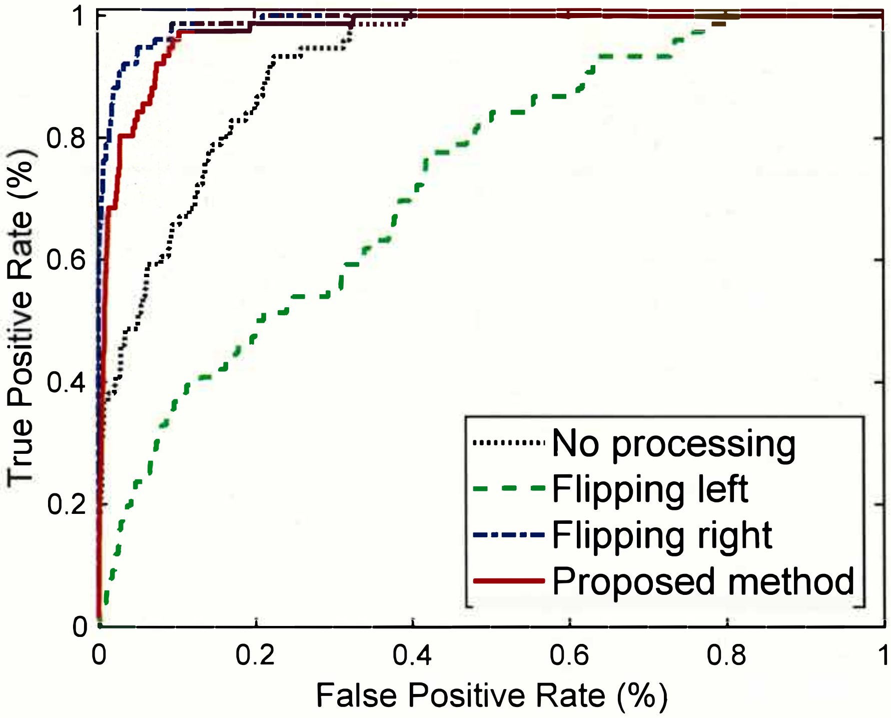


Figure 4.13: The performance comparison of four methods when the face is 70% occluded.

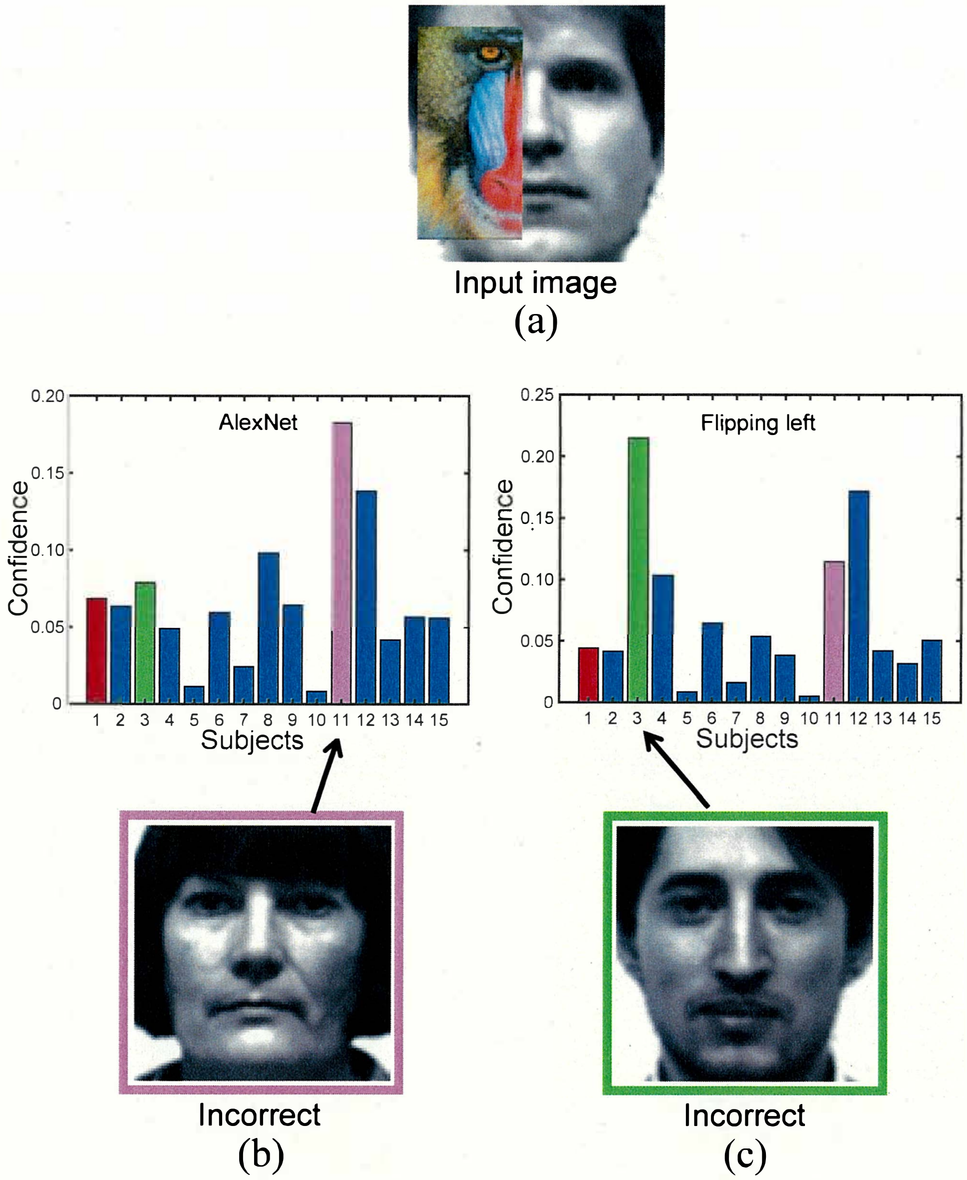


Figure 4.14: An occluded face classification example predicted by the first two methods. (a) Input image; (b) the result of the original classifier recognition of AlexNet; (c) the result of recognition after flipping left. Confidence refers to the prediction score of a test image belonging to a certain identity.

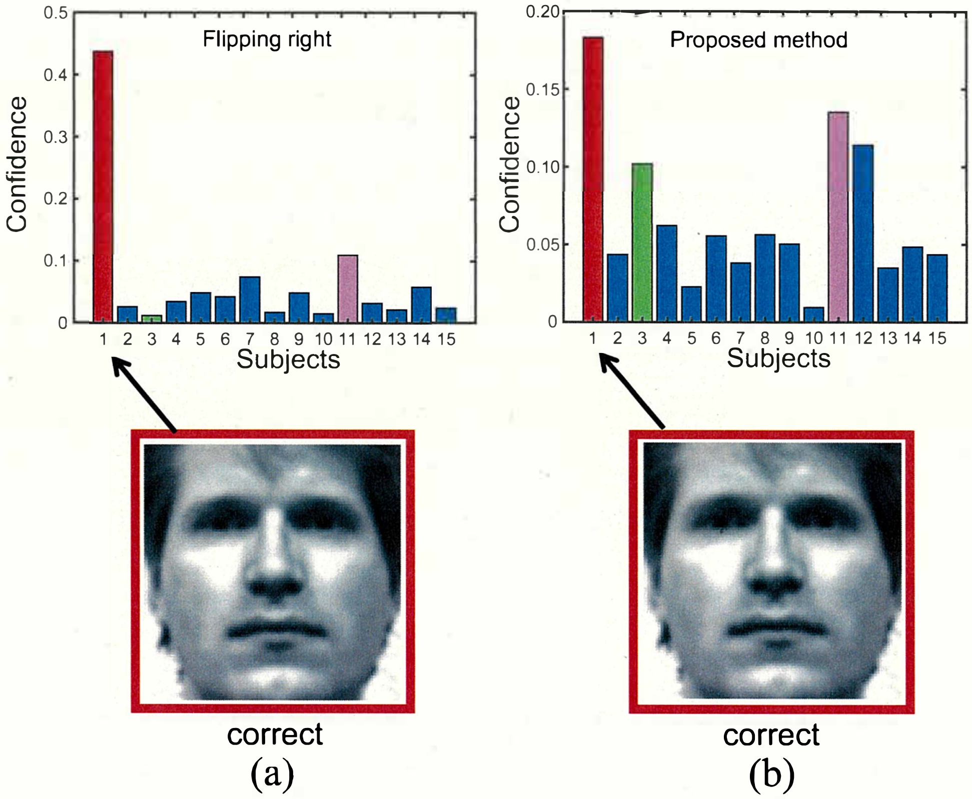


Figure 4.15: An example of classification result predicted by the other two methods. (a) the result after flipping right; (b) the result that sums the no-processing, flipping left, and flipping right. Confidence refers to the prediction score of a test image belonging to a certain identity.

experiment, we evaluated the recognition rate of each group of occluded faces, as shown in Table 4.6. Here, we introduce two methods to combine no processing, flipping left and flipping right together. Average score refers to the average score of the aforementioned three methods. Proposed method refers to the maximum score of the three methods. Specifically, let $p = (s_1, s_2, s_3, ...s_k)$ denote the score of a test image by the classifier, s_i denote the possibility of the test image belonging to person number i. In our case, the number of people k equals to 15. The average score is denoted by the following formula:

$$p_1 = (p_a + p_l + p_r)/3, \tag{4.4}$$

where p_a refers to the score of no processing, p_l refers to the score of flipping left, p_r refers to the score of flipping right. The proposed method is denoted by the following formula:

$$p_2 = \begin{cases} p_l \; max(p_l) > max(p_a) \quad \& \quad max(p_l) > max(p_r) \\ p_r \; max(p_r) > max(p_l) \quad \& \quad max(p_r) > max(p_a) \\ p_a \qquad\qquad\qquad otherwise \end{cases} \cdot \tag{4.5}$$

From the results of Table 4.6, the following conclusions can be drawn:

1. In the result of “no processing”, the order of importance is: $mouth > eye > nose$.

2. In the result of “average score”, the order of importance is: $nose = mouth > eye$.

3. In the result of “proposed method”, the order of importance is: $mouth > nose > eye$.

4. “Average score” eliminates the bad effects of eye occlusion, however, the improvement is not obvious for nose and mouth.

5. The proposed method with maximum score performs better than the average score, which means the method of choosing the maximum score is better than the method of averaging scores.

The reason why the proposed method is good for eye occlusion is that the occluded eye is in the half part face, and the bad effect of occlusion is eliminated after flipping. Since proposed method performed better than the average score, this paper uses it as our proposed method.

To validate the performance of reconstructing occluded faces, classification results of four reconstruction strategies on the 70% occluded face images are illustrated by receiver operating characteristic (ROC) curves as in Fig. 4.13. The curve of no processing represents the recognition accuracy obtained by inputting the original image without any processing, and flipping left refers to the score that uses the image obtained by flipping the left-half face to the right and covering the original right side as the test image. Flipping right refers to the score that use the image obtained by flipping the right-half face to the left and covering the original left side as the test image. The proposed method refers to the strategy of Eq. 3.24. Thus, from the ROC curves, we can see that our method can achieve better performance than the method that without any post-processing. The main advantage of our proposed method is that we do not need to consider which side has the bad occlusion condition. The result obtained by calculating the maximum recognition score of the no processing, flipping left, and flipping right can lead to a correct recognition.

To verify why our method can achieve good performance, we examined the score distribution for each strategy as in Fig. 4.14 and Fig. 4.15. In the histogram, red color represents the subject 1, green represents the subject 3, and purple represents the subject 11. The rectangular box on the output image below the histogram also uses the same color to indicate which subject does the output image belongs to. We input the image (a) as the test image. Using the normal method that input the original image to the classifier got from training process. In Fig. 4.14, (b) is the face recognition result of AlexNet without processing, where No.11 subject reaches the highest confidence. However, the result is incorrect. The result of flipping left method is (c), where No.3 subject reaches the highest confidence. The result of flipping left method is incorrect. As for the flipping right method, the face recognition result is shown in Fig. 4.15(a), where No.1 subject reaches the highest confidence. The result of flipping right method is correct. The result of proposed method is Fig. 4.15(b), where No.1 subject reaches the highest confidence. The result of proposed method is correct. Flipping right method got the correct recognition result, but the highest recognition score is performed by our proposed method. From the face image, we can easily know the fact that in the test face image, the left-half face is occluded. Thus, if we flipped the right-half face to the left and cover the original left face to generate a conjecture face, the negative effect of occlusion on the left face will be reduced in the conjecture face. Our proposed method uses the flipping strategy to post-process the face image. The combination of CNN and flipping strategy can greatly improve the accuracy of face recognition that under the influence of occlusion.

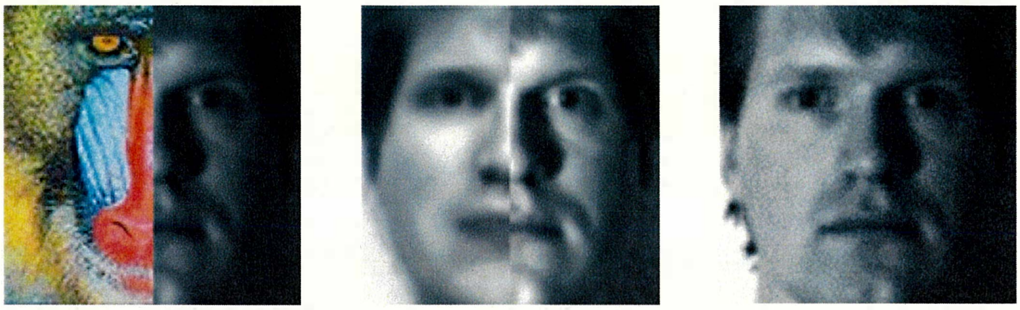


Figure 4.16: The result of the reconstructed images using linear combination. From left to right: target occluded face, reconstructed face, and unoccluded face.

As shown in Fig. 4.16, the comparison of reconstructed image and the real image is given. From left to right, the images show the target occluded face, reconstructed face (calculated by the coefficient matrix W of Section 3.2), and the real face. This comparison showed that our reconstruction results of the occluded regions are similar to the real images. The reconstructed region in Fig. 4.16 is dim and blurred compared to the real image. This is because the reconstruction makes use of the statistical information of the right half faces and there is still a reconstruction error after many iterations. Figure 4.17 shows the recognition results of the reconstruction of linear combination and flipping scheme. The recognition results of linear combination and flipping scheme are shown as Fig. 4.17 (a) and (b). Both face recognition accuracy results are exactly the same. The face recognition confidence scores of all subjects in the dataset have no significant difference and this does not change the final prediction result.

In the case where the middle part of face is occluded by the mask, an orthogonalized coupled learning model [65] can be learned to approximate no-mask face images with images in the masked face database. Then, this model can be used to

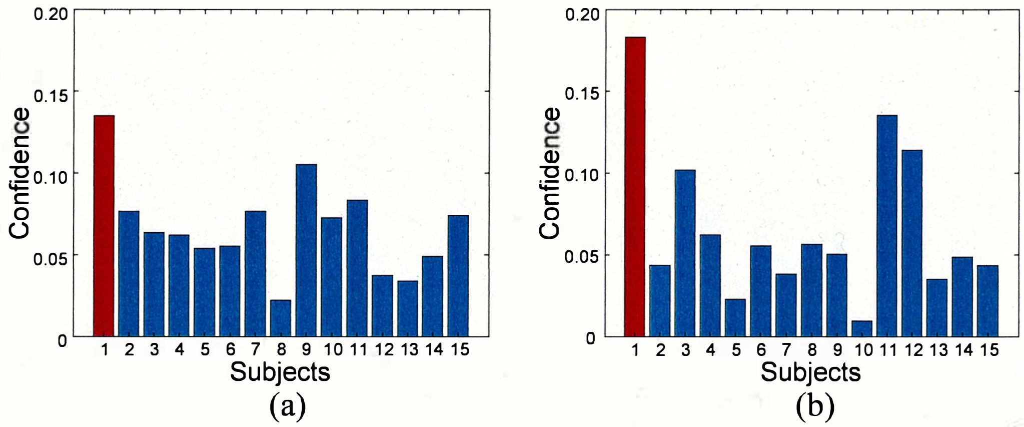


Figure 4.17: Examples of the recognition results using linear combination (a) and flipping scheme (b) about subject 1. The red bar represents the ground truth.

predict the no-mask face image of a person with masked face images of the same person. The prediction results may differ by the number and quality of images in the database. We plan to combine the orthogonalized coupled learning model with the aforementioned method to handle more cases of occlusions in the future.

4.5 Visualization of Facial Latent Spaces

The variational autoencoder model described in Section 3.5 has two parts: the encoder and the decoder. The encoder takes an image input and outputs a compressed representation (the encoding), which is a vector of pre-defined length, which equals to 100 for the whole face images in this paper. The decoder takes the compressed representation, decodes it, and recreates the original image. In this experiment, a six-layer encoder network is built as shown in Table. 4.7, and an eight-layer decoder network is built as shown in Table. 4.8. In these tables, each layer is named with the first few characters of the layer type and the number of times that each type of layer appeared. The filter size is denoted by two positive integers which represent the height and width of the filter respectively. The stride represents the step size for traversing input. The padding represents the input edge padding, where the value of "same" represents the method of zero-padding. The method of zero-padding adds zeros to the edge of the input image and makes the output image to have the same size of the input image. For instance, suppose the input image has the size of 10×10, the stride is set as $(2, 2)$, and the padding is set as "same", then the zero-padding method firstly adds zeros and makes the input image to be 12×12. The size of the output image will be $(12 - 3 + 1)/2 = 5$. When the padding is set as "same", the calculation of the output image size can be written as follows:

$$w_{out} = \text{ceil}\left(\left(w_{in} - w_{filter} + 1\right) / s_{stride}\right) \tag{4.6}$$

where w_{out} and w_{in} denote the width of the output image and that of the input image respectively. The width of the filter and the stride are represented by w_{filter} and s_{stride} respectively. The function of "ceil" rounds the input value to the nearest integer greater than or equal to that value. The height can be calculated in the same way as the width. Actually, the width of the input image or the filter is usually equivalent to the height.

To make calculations more numerically stable, the range of possible values for the desired distribution is increased from $[0, 1]$ to $[-\infty, 0]$ by making the network learn from the logarithm of the variances. The means μ and the variances σ (the logarithm is taken while calculation) are the two vectors to create the distribution to sample from. As shown in Fig. 3.14, multiple 2D convolution layers followed by a fully connected layer are used as the encoder to downsample from the $32 \times$

Table 4.7: The encoder network used to visualize the latent space of faces.

No.	Layer Name	Layer Type	Filter Size	# Filters	Stride	Padding	Layer Description
1	input_encoder	Image Input	–	–	–	–	Input images of $32 \times 32 \times 1$
2	conv1	Convolution	3×3	32	$(2, 2)$	"same"	The first convolutional layer
3	relu1	ReLU	–	–	–	–	The first rectified linear unit
4	conv2	Convolution	3×3	64	$(2, 2)$	"same"	The second convolutional layer
5	relu2	ReLU	–	–	–	–	The second rectified linear unit
6	fc_encoder	Fully connected	–	–	–	–	the fully connected layer

Table 4.8: The decoder network used to visualize the latent space of faces.

No.	Layer Name	Layer Type	Filter Size	# Filters	Stride	cropping	Layer Description
1	input	Latent variable input	–	–	–	–	The latent variable of $1 \times 1 \times 100$
2	transpose1	Transposed convolution	8×8	64	$(8, 8)$	"same"	The first transposed convolutional layer
3	relu1	ReLU	–	–	–	–	The first rectified linear unit
4	transpose2	Transposed convolution	3×3	64	$(2, 2)$	"same"	The second transposed convolutional layer
5	relu2	ReLU	–	–	–	–	The second rectified linear unit
6	transpose3	Transposed convolution	3×3	32	$(2, 2)$	"same"	The third transposed convolutional layer
7	relu3	ReLU	–	–	–	–	The third rectified linear unit
8	transpose4	Transposed convolution	3×3	1	$(1, 1)$	"same"	The fourth transposed convolutional layer

16 × 1 half face image to the 1 × 1 × 50 encoding of the latent space. Then, three transposed 2D convolution layers are used to scale up the encoding back into a 32 × 16 × 1 reconstructed half face image. Another instance of whole face image 32 × 32 × 1 is also trained for the encoding of 1 × 1 × 100. To compare the difference between whole face and half face latent spaces, t-Distributed Stochastic Neighbor Embedding (tSNE) is utilized to reduce the dimensionality for the 2D data point distribution visualization.

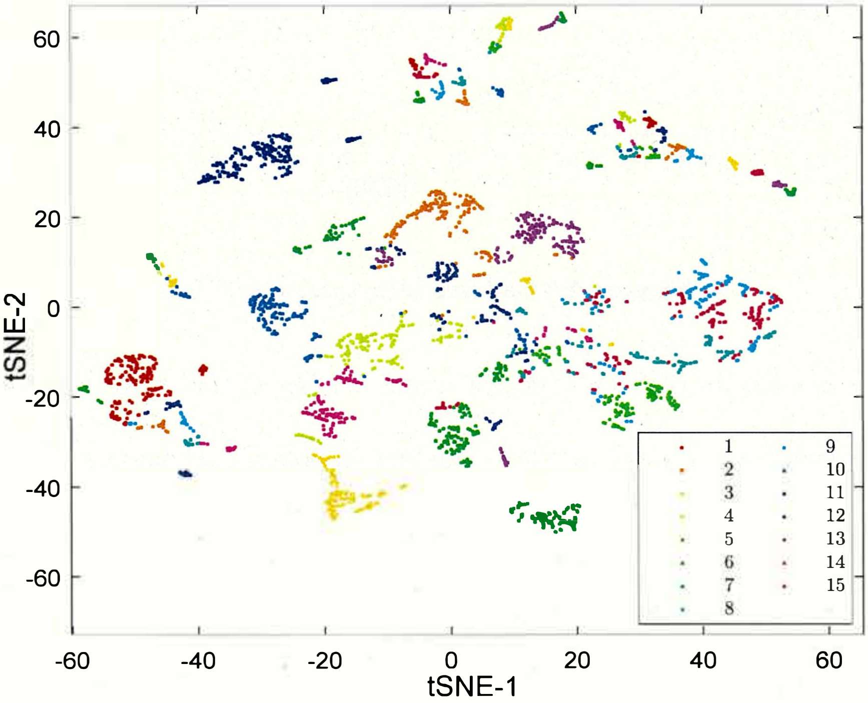


Figure 4.18: Whole face latent space visualization.

To enhance the performance of latent space visualization, 2970 images were augmented from the Yale dataset [62] by translation, flip, crop, and scale. 30% of the augmented images are used for validation and 70% for training. In the experiment, the transfer learning of pre-trained CNNs applied stochastic gradient descent with momentum with a mini-batch size of 20, an L2 regulation factor of 1×10^{-4},

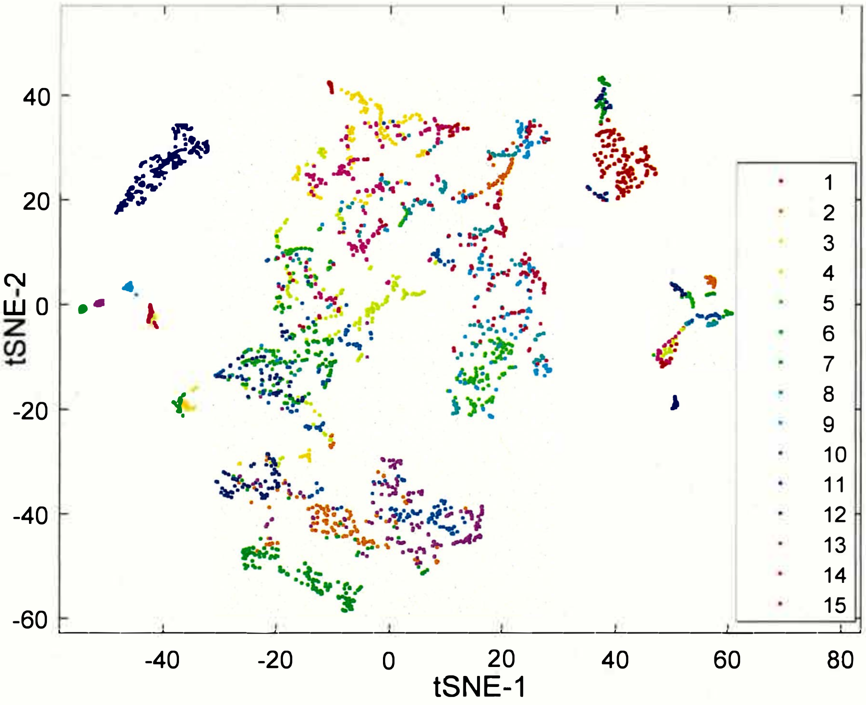


Figure 4.19: Half face latent space visualization.

and a momentum of 0.95. The maximum number of epoch was set to 6. The initial learning rate was 1×10^{-4}. Training on both the whole-face and half-face images was performed with the same parameter configuration for comparison. Six pretrained CNN architectures, i.e., AlexNet, GoogLeNet, SqueezeNet, ResNet-50, Inception-v3, and DenseNet-201 were used in transfer learning with both strategies to perform facial image classification on the two strictly cropped datasets. Evaluation metrics $accuracy$, $sensitivity$, $specificity$, and $AUCscore$ were used in this study for all the testing images. The PC used for the experiment had a CPU of i7-8700 (3.19 GHz), RAM of 16 GB, and GPU of GeForce GTX 1650. The results for these evaluation metrics are presented in Table 4.1. As shown in the table, all six pretrained CNN models were evaluated with similar input image sizes. The second column shows the number of layers in the source code level. Each model was evaluated using both the whole-face and half-face strategies. By comparing the results of two different strategies, it can be concluded that the half-face training strategy outperformed the whole-face training strategy on AlexNet, SqueezeNet, GoogLeNet, and Inception-v3 in terms of the $accuracy$, $sensitivity$, $specificity$, and $AUCscore$. The ResNet-50 and DenseNet-201 demonstrated a slightly worse performance on the half-faces than on the whole face.

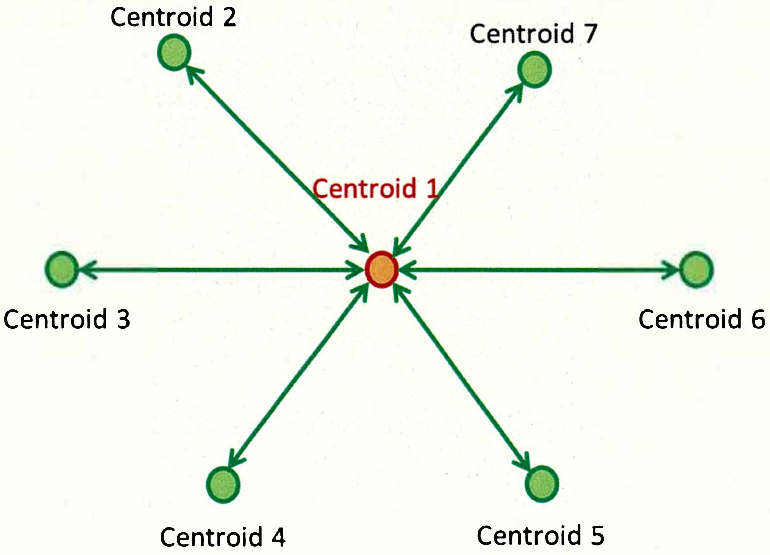


Figure 4.20: Illustration of calculating the the average distance between the centroid 1 and all the other centroids.

Two different variational autoencoder networks are trained with the maximum epoch of 100, initial learning rate of 0.001 for the "adam optimizer", and mini-batch size of 512. One is learned for the latent space for whole face training set, and the other is for half face training set, which randomly select the left half face or the flipped right half face. As shown in Fig. 4.18 and Fig. 4.19, the dimension of the latent space vector is reduced to 2 for visualization of the separative ability comparison. Result of dimensionality reduction shows the latent variables of the

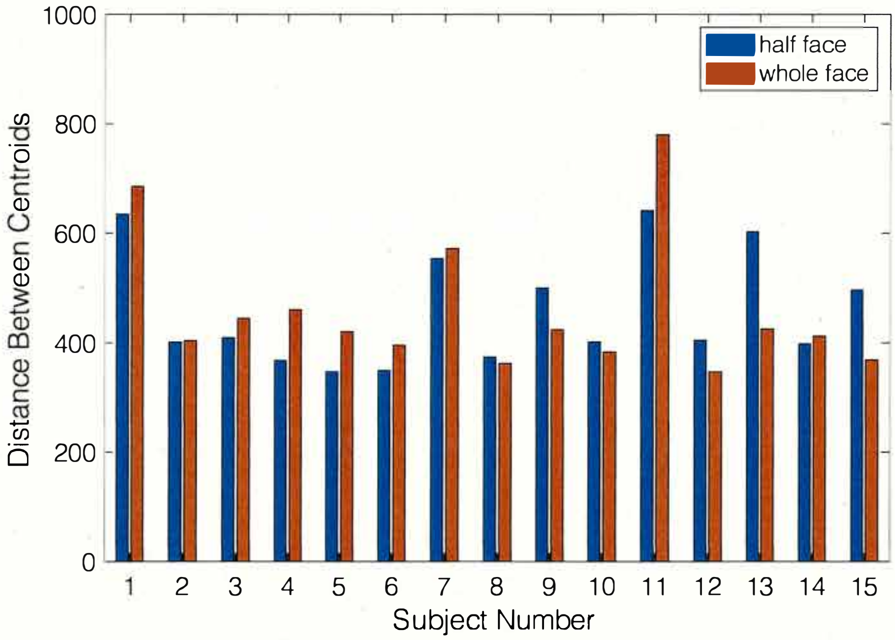


Figure 4.21: The comparison of cluster distributions between the whole faces and half faces.

whole face or half face encoder network. Each point indicates a face sample image. Each color indicates a person identity. Figure 4.18 illustrates the distribution of all 15-class whole face sample images. Figure 4.19 illustrates the distribution of all 15-class half face sample images. By comparison, the blue points representing person id of 11 form the most isolated cluster in both figures. Most of the other points of the same color are located near to each other in both figures. We also conducted the experiments to evaluate the distribution of color points, as showed in Fig. 4.21. The x-axis is the subject number of the 15-class images and the y-axis is the average distance between the current centroid and other centroids, which can be written as follows:

$$\varphi(k) = \frac{\sum_{i=1}^{n} \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}}{n-1}, i \neq k, k = 1, 2, \cdots n \tag{4.7}$$

where $\varphi(k)$ denotes the average distance between the centroid k and all the other centroids. The coordinates of centroid k is denoted by (x_k, y_k). The average distance is the average of the distance from each individual to all the other individuals, as shown in Fig. 4.20. Here, center point of the color points, which is also called centroid, is used to do this experiment. A large value of $\varphi(k)$ represents centroid k is far away from the others, which also means the sample images of individual k can be easily separated from others and the classifier has a low possibility to make mistakes on the sample images of individual k. From the visualization of latent space and the evaluation of the color points distribution for whole face and half face, we can conclude that they contain the latent encoding variables of similar separative ability in general. As shown in Fig. 4.21, for subject 11, the whole face is better than the half face in the separative ability of the latent space. Influenced by the bangs, the whole face has more distinct recognition features than half face. However, for subject 13 and subject 15, the half face is better than the whole face in the separative ability of the latent space. For subject 13 and subject 15, influenced by the glasses and their reflections, the recognition of the whole face is greatly affected. Instead, performing the recognition task with half face has less noisy.

Besides, the dimension of the latent space may have an influence on the separative ability comparison between whole faces and half faces. Another similar experiment is designed to verify this statement, where the dimension of the latent space is reduced to 10, and the case of half faces is divided into two cases of the left half face and the right half face. The learning parameters keep the same configuration as the previous experiment and the network structure also does not change in order to show the influence of a low dimension latent space. As shown in Fig. 4.22 and Fig. 4.23,

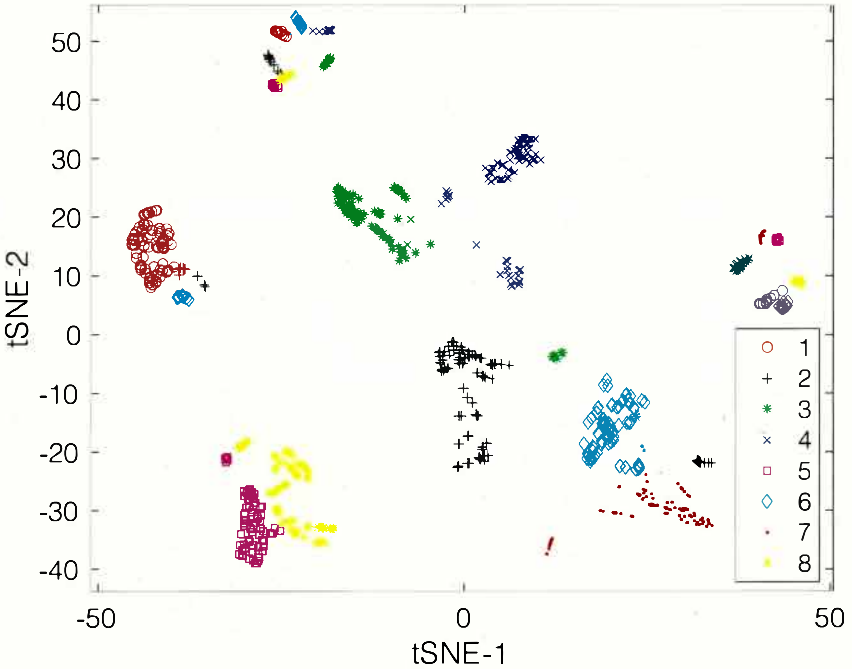


Figure 4.22: Whole face latent space visualization for subject 1 to subject 8 when the dimension of the latent space is reduced to 10.

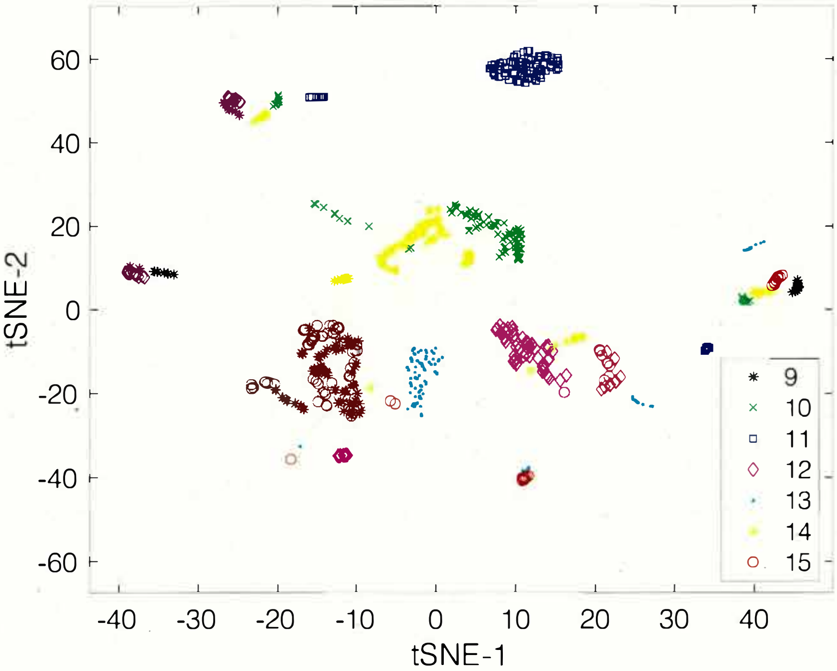


Figure 4.23: Whole face latent space visualization for subject 9 to subject 15 when the dimension of the latent space is reduced to 10.

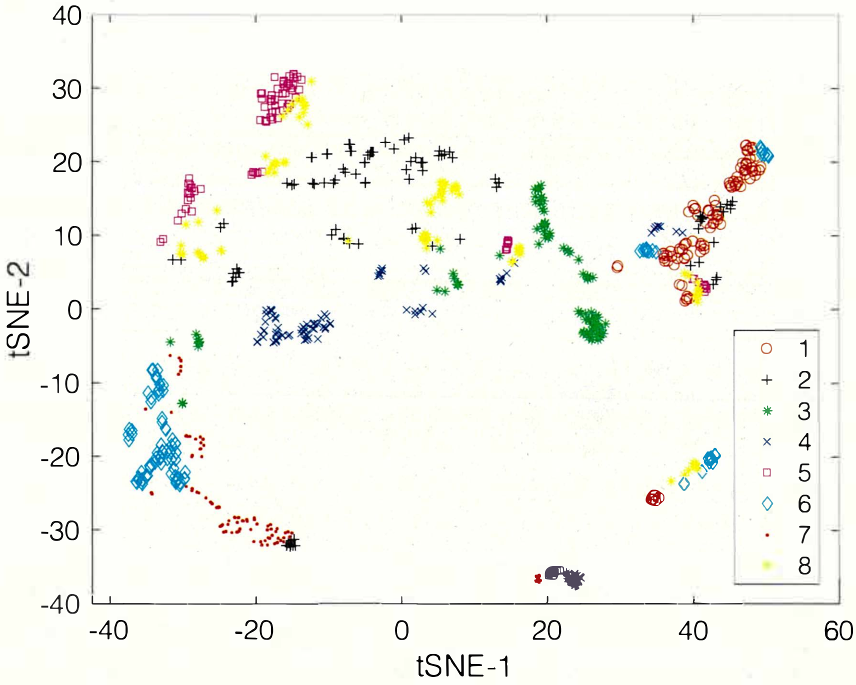


Figure 4.24: Left half face latent space visualization for subject 1 to subject 8 when the dimension of the latent space is reduced to 10.

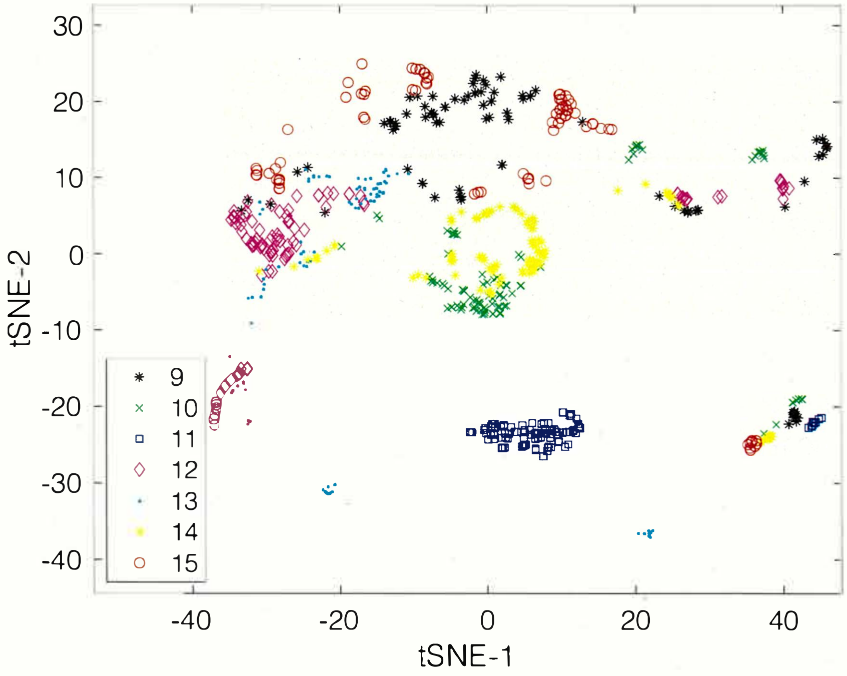


Figure 4.25: Left half face latent space visualization for subject 9 to subject 15 when the dimension of the latent space is reduced to 10.

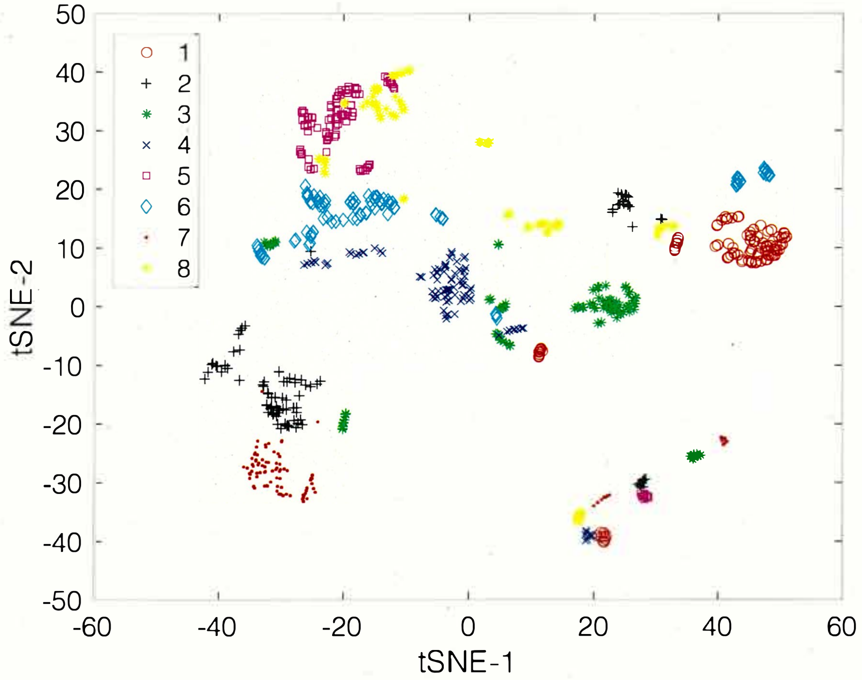


Figure 4.26: Right half face latent space visualization for subject 1 to subject 8 when the dimension of the latent space is reduced to 10.

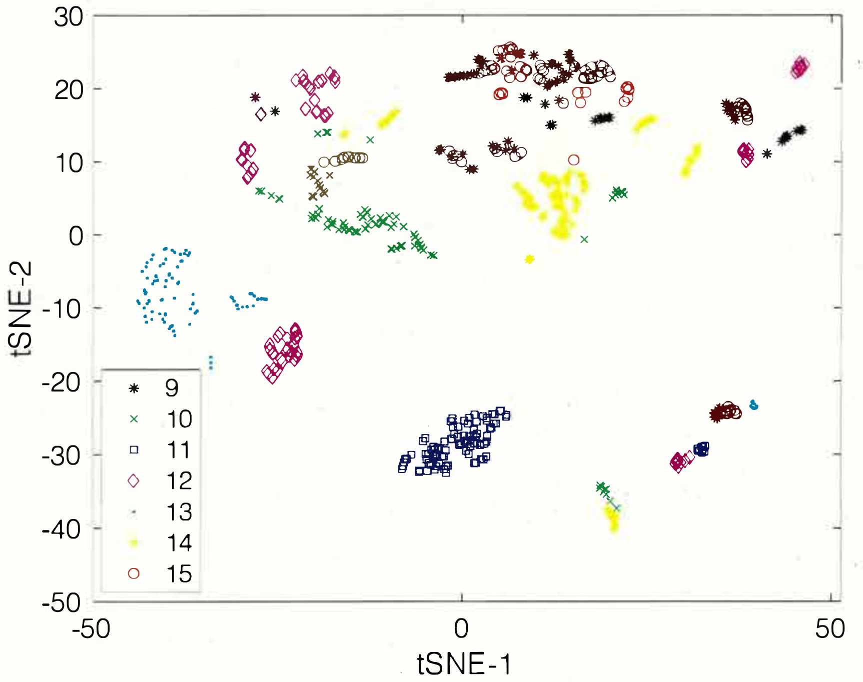


Figure 4.27: Right half face latent space visualization for subject 9 to subject 15 when the dimension of the latent space is reduced to 10.

the whole face image is encoded into a 10-dimensional latent space and visualized in a two-dimensional t-SNE scatter figure for the first 8 subjects and another for the rest subjects. As shown in Fig. 4.24 and Fig. 4.25, the left half face image is also encoded into a 10-dimensional latent space and visualized in the two-dimensional t-SNE scatter figure for the first 8 subjects and another for the rest subjects. As shown in Fig. 4.26 and Fig. 4.27, the right half face image is also encoded into a 10-dimensional latent space and visualized in the two-dimensional t-SNE scatter figure for the first 8 subjects and another for the rest subjects. By comparison, both the whole face and two half faces show the latent space visualization with a clear boundary for subject 9. For other subjects, it is difficult to distinguish one from another because the colored points do not display a clear boundary. Thus, the average distance between the current centroid and other centroids is also used to show a quantitative indicator for the difference, as shown in Fig. 4.28. By comparison, the whole face has a slightly larger distance value than the left half face and the right half face on subject $3, 4, 7, 8, 10$, and 11. It represents that on these subjects, the whole face latent space has a better separative ability. However, on subject $2, 9$, and 13, the right half face latent space has a better separative ability. By decoding the latent space, the input face images can be reconstructed to make a face-like image, as shown in Fig. 4.29 for whole face input images, Fig. 4.30 for left half face input images, and Fig. 4.31 for right half face input images.

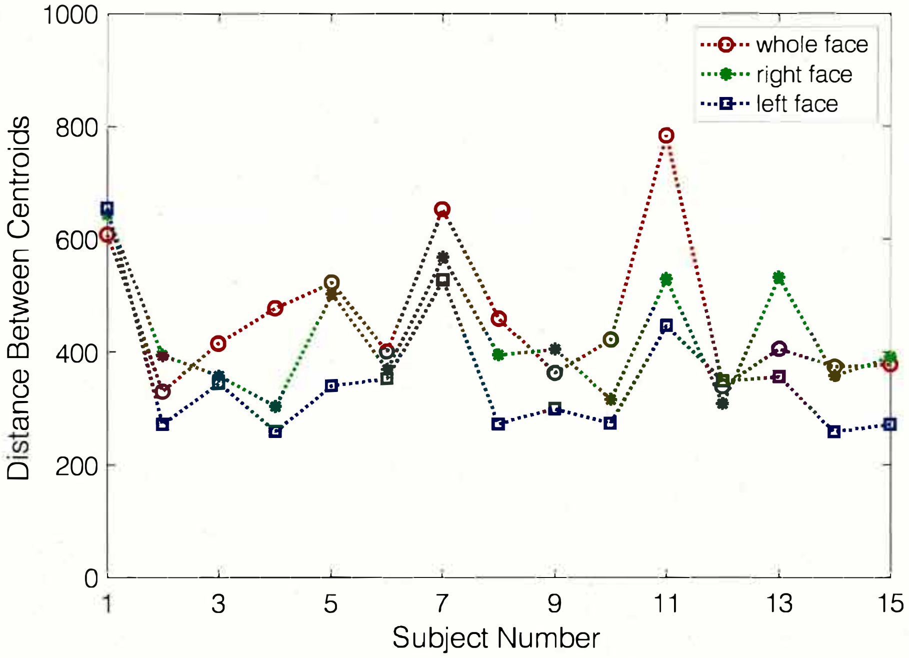


Figure 4.28: The comparison of cluster distributions between the whole face, left half face, and right half face.

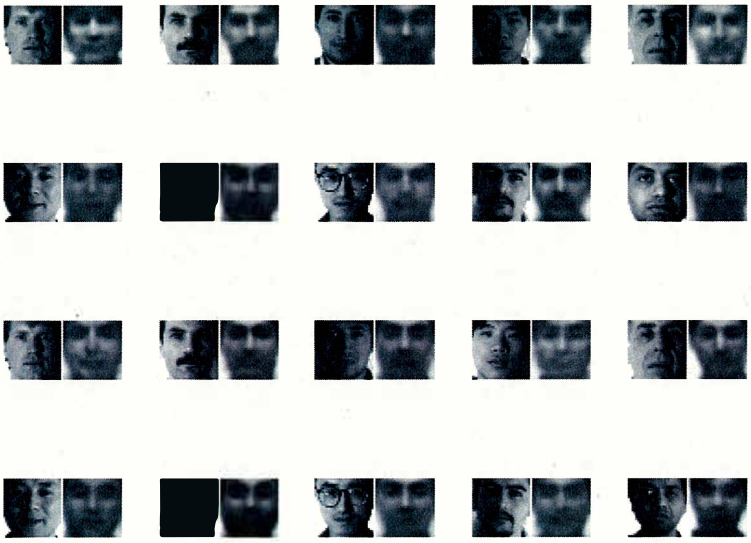


Figure 4.29: The comparison between original whole face images and reconstructed whole face images.

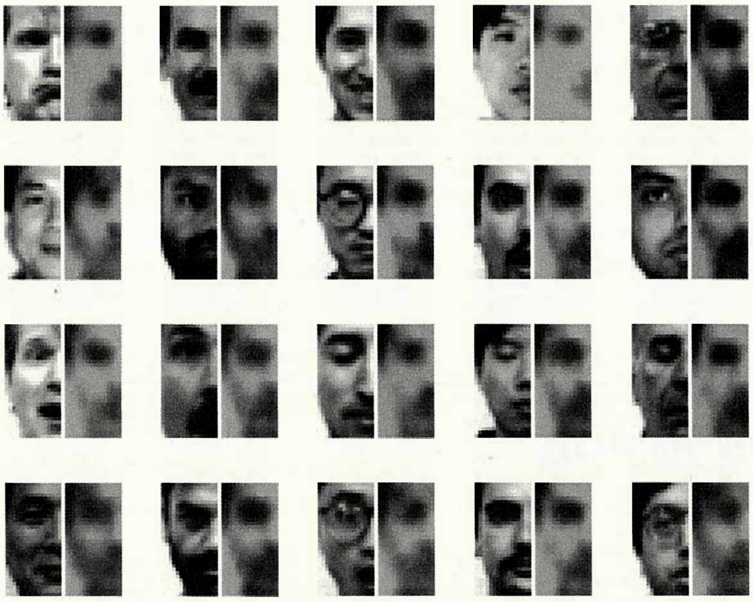


Figure 4.30: The comparison between original left half face images and reconstructed left half face images.

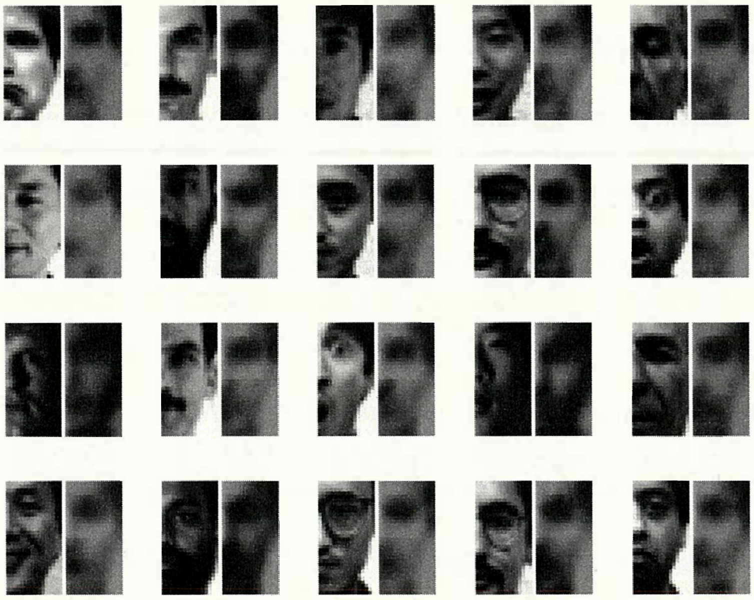


Figure 4.31: The comparison between original right half face images and reconstructed right half face images.

Chapter 5

Conclusion

Most face detection and recognition tasks are based on the training of intact facial images and corresponding labels. Both the three-dimensional structure and two-dimensional appearance from the frontal view of human faces are approximately bilaterally symmetrical in general. However, sometimes, illumination on the left half face and the right-half face is uneven. In this case, the symmetrical characteristic of human faces can facilitate expressing distinct identity information. This is because even if one side of facial image is corrupted by noise, the opposite side can still be used for feature extraction. The recent literature indicates that face recognition and facial expression classification has achieved a high accuracy on benchmark datasets with a large number of face images in the wild. However, unlike the purpose of recognizing as many people as possible, real applications for families or companies usually aim to recognize a small group of people as accurate as possible. In case of the face is partially occluded, convolutional solutions always simply put images with occlusions into the training dataset and hope the convolution neural network learn a model robust to partial occlusion. These processes not only increase the burden of learning, but also affect the model to identify normal images without occlusions.

To address this problem, an automatic selection of the better half of the face can be used for identity recognition with only a single half face. Different from the MegaFace challenge of recognizing millions of identities in the wild, this thesis focuses on building recognition systems for a small number of people with fewer training images, for example, building access control systems for research laboratory members or family members. This thesis proposes an artificial face image construction method and a half-face training strategy for transfer learning of pre-trained conventional neural network models. The facial image reconstruction to discard the influence of partial occlusion is also discussed. Based on the phenomenon that human faces are roughly symmetrical, the intact half face can be used to reconstruct the facial information of the occluded areas. Specifically, occlusion on the left-half face is reconstructed with a linear combination of features on the right-half face, and vice versa. The process is modeled by keeping row sparsity for the coefficient matrix with $l_{2,1}$-norm regularization while minimizing the reconstruction error. An alternative iterative algorithm is proposed to solve the optimization problem. To validate the effectiveness of the reconstruction, the pre-trained CNN model is trained on normal face images and tested with various occluded images. Extensive experimental results show that the proposed method improves the performance of state-of-the-art models by utilizing the symmetrical characteristics of human faces.

However, in the case where the middle part of face is occluded by the mask, an orthogonalized coupled learning model can be learned to approximate no-mask face images with images in the masked face database. Then, this model can be used to predict the no-mask face image of a person with masked face images of the same person. The prediction results may differ by the number and quality of images in the database. I plan to combine the orthogonalized coupled learning model with the aforementioned method in this thesis to handle more cases of occlusions in the future.

Bibliography

[1] D. S. Trigueros, L. Meng, and M. Hartnett, “Face recognition: From traditional to deep learning methods,” *arXiv preprint arXiv:1811.00116*, 2018.

[2] H. B. Fredj, S. Bouguezzi, and C. Souani, “Face recognition in unconstrained environment with CNN,” *Vis. Comput.*, vol. 37, no. 2, pp. 217–226, 2021.

[3] B. Zhang, B. Tondi, and M. Barni, “Adversarial examples for replay attacks against cnn-based face recognition with anti-spoofing capability,” *Comput. Vis. Image Underst.*, vol. 197-198, p. 102988, 2020.

[4] M. D. Kelly, *Visual identification of people by computer.* Department of Computer Science, Stanford University., 1970, no. 130.

[5] T. Kanade, “Picture processing by computer complex and recognition of human faces,” *Ph. D. Thesis, Kyoto University*, 1973.

[6] K. Delac and M. Grgic, “A survey of biometric recognition methods,” in *Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine.* IEEE, 2004, pp. 184–193.

[7] M. Wang and W. Deng, “Deep face recognition: A survey,” *Neurocomputing*, vol. 429, pp. 215–244, 2021.

[8] Y. Kortli, M. Jridi, A. Alfalou, and M. Atri, “Face recognition systems: A survey,” *Sensors*, vol. 20, no. 2, p. 342, 2020.

[9] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *European conference on computer vision.* Springer, 2016, pp. 499–515.

[10] R. M. Ramadan and R. F. Abdel-Kader, “Face recognition using particle swarm optimization-based selected features,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 2, pp. 51–65, 2009.

[11] B. Heisele, P. Ho, and T. Poggio, “Face recognition with support vector machines: Global versus component-based approach,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2. IEEE, 2001, pp. 688–694.

[12] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, “Boosting local binary pattern (lbp)-based face recognition,” in *Chinese Conference on Biometric Recognition.* Springer, 2004, pp. 179–186.

[13] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, “Face recognition using laplacianfaces,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 328–340, 2005.

[14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[15] X. Tan and B. Triggs, "Fusing gabor and lbp feature sets for kernel-based face recognition," in *International workshop on analysis and modeling of faces and gestures.* Springer, 2007, pp. 235–249.

[16] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[17] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *European conference on computer vision.* Springer, 2012, pp. 566–579.

[18] Y. Sun, 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, pp. 1891–1898.

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[20] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning and disentangling face representation by multi-view perceptron," in *Proc. NIPS*, 2014.

[21] Y. Liu, J. Chen, and Y. Qiu, "Joint multi-patch and multi-task cnns for robust face recognition," *IEICE Trans. Inf. Syst.*, vol. 103-D, no. 10, pp. 2178–2187, 2020.

[22] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[23] M. Mehdipour Ghazi and H. Kemal Ekenel, "A comprehensive analysis of deep learning based representation for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 34–41.

[24] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma, "Face recognition with contiguous occlusion using markov random fields," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1050–1057.

[25] M. Zou, M. You, and T. Akashi, "Application of facial symmetrical characteristic to transfer learning," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 16, no. 1, pp. 108–116, 2021.

[26] ——, "Reconstruction of partially occluded facial image for classification," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 16, no. 4, pp. 600–608, 2021.

[27] W. Zhang, X. Zhao, J.-M. Morvan, and L. Chen, "Improving shadow suppression for illumination robust face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 611–624, 2018.

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[29] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive his-

togram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[30] S. Shan, W. Gao, B. Cao, and D. Zhao, "Illumination normalization for robust face recognition against varying lighting conditions," in *2003 IEEE International SOI Conference. Proceedings (Cat. No. 03CH37443).* IEEE, 2003, pp. 157–164.

[31] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[32] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition.* IEEE, 2002, pp. 53–58.

[33] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[34] R. Ishiyama, M. Hamanaka, and S. Sakamoto, "An appearance model constructed on 3-d surface for robust face recognition against pose and illumination variations," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 3, pp. 326–334, 2005.

[35] ——, "Face recognition under variable pose and illumination conditions using 3d facial appearance models," *Systems and Computers in Japan*, vol. 38, no. 2, pp. 57–70, 2007.

[36] L. Song, D. Gong, Z. Li, C. Liu, and W. Liu, "Occlusion robust face recognition based on mask learning with pairwise differential siamese network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 773–782.

[37] T. Y. Kim, K. M. Lee, S. U. Lee, and C.-H. Yim, "Occlusion invariant face recognition using two-dimensional pca," in *Advances in Computer Graphics and Computer Vision*. Springer, 2007, pp. 305–315.

[38] H. Jia and A. M. Martinez, "Support vector machines in face recognition with occlusions," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 136–141.

[39] B. Buckley and M. Hunter, "Say cheese! privacy and facial recognition," *Computer Law & Security Review*, vol. 27, no. 6, pp. 637–640, 2011.

[40] C. Garvie, A. Bedoya, and J. Frankle, "The perpetual line-up. unregulated police face recognition in america. georgetown law center on privacy & technology," 2019.

[41] T. Ring, "Privacy in peril: is facial recognition going too far too fast," *Biometric Technology Today*, vol. 2016, no. 7-8, pp. 7–11, 2016.

[42] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, p. 1188, 2020.

[43] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[44] L. Zhao and Y.-H. Yang, "Theoretical analysis of illumination in pca-based vision systems," *Pattern recognition*, vol. 32, no. 4, pp. 547–564, 1999.

[45] F. Y. Shih and C.-F. Chuang, "Automatic extraction of head and face boundaries and facial features," *Information Sciences*, vol. 158, pp. 117–130, 2004.

[46] D. Kumar, J. Garain, D. R. Kisku, J. K. Sing, and P. Gupta, "Unconstrained and constrained face recognition using dense local descriptor with ensemble framework," *Neurocomputing*, vol. 408, pp. 273–284, 2020.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[50] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[51] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[52] Y. Takano, "Why does a mirror image look left-right reversed? a hypothesis of multiple processes," *Psychonomic bulletin & review*, vol. 5, no. 1, pp. 37–55, 1998.

[53] M. You and T. Akashi, "Multi-view face detection using flipping scheme," in *Machine Vision and Image Processing Conference IMVIP 2014*, 2014.

[54] Y. Xu, Z. Zhang, G. Lu, and J. Yang, "Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification," *Pattern Recognition*, vol. 54, pp. 68–82, 2016.

[55] Y. Moses, Y. Adini, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," in *European conference on computer vision*. Springer, 1994, pp. 286–296.

[56] M. Elawady, C. Ducottet, O. Alata, C. Barat, and P. Colantoni, "Wavelet-based reflection symmetry detection via textural and color histograms," *CoRR*, 2017.

[57] Y. Takano and A. Tanaka, "Mirror reversal: Empirical tests of competing accounts," *Quarterly journal of experimental psychology (2006)*, vol. 60, pp. 1555–84, 12 2007.

[58] M. You and T. Akashi, “Multi-view face detection using frontal face detector,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 7, pp. 1011–1019, 2018.

[59] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[60] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *In Proceeding of the International Conference on Learning Representations (ICLR)*, 2014.

[61] M. Zou and A. Takuya, “Latent space visualization of half face and full face by generative model,” *International Society for Optics and Photonics*, pp. 153–158, 2021.

[62] S. Fischer, R. Klinkenberg, I. Mierswa, and O. Ritthoff, “Yale: Yet another learning environment–tutorial,” *Colloborative Research Center*, vol. 531, 2002.

[63] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” *CoRR*, vol. abs/1411.7923, 2014.

[64] K.-C. Lee, J. Ho, and D. J. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[65] T. Nagata, “Orthogonalized coupled learning and application for face image processing,” in *2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics*. IEEE, 2018, pp. 1–7.