FACE RECOGNITION USING CONVOLUTIONAL MACROPIXEL COMPARISON APPROACH

by

Yunke Li

B.Sc. Electronic and Information Engineering
Tianjin University, 2013

THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

April, 2018

Abstract

Convolutional Neural Network (CNN) is a widely used deep learning framework and is applied in the field of face recognition achieving outstanding results. Macropixel Comparison Approach is a shallow mathematical approach that recognizes face by comparing original pixel blocks of face images. In this thesis, we are inspired by ideas of the currently popular deep neural network framework and introduce two features into the mathematical approach: deep overlap and weighted filter. The aim is exploring if the idea of deep learning could benefit mathematical method which might extends the scope of face recognition research. Results from our experiments show that the new proposed approach achives markedly better recognition rates than the original macropixel method.

Contents

List of Figures

Acknowledgements

I want to specifically thank my supervisor, Dr. Liang Chen. Thank him for giving me great advices, support, supervision, understandment, patience and help in my study and life. He will always be my example in the future. This thesis can not be done without him.

I also want to thank University of Northern British Columbia for the learning and life support. Thanks to Prof. Jernej Polajnar, Prof. Desanka Polajnar and Yan Ling for their support when I'm doing TA. The financial support from mitacs and TA program through Dr. Chen's go-between also helps me a lot.

To all my colleagues in UNBC, Negar, Raj, Farhana, Meng Xi, Tony Zhuang, Olivia Wang and Hongyuan Shi, thank you for the help and support in my research and study.

To all the friends I met in Prince George, Boshi, Finch, Huan, Michelle, Jess, Ricki, Danny, Steven, Jarratt, thank you for the company, support and help.

To Andrea Li, my girlfriend, my life in Canada would not be this colorful and happy without you.

I'm very grateful to my parents. Thank you for your guidance and love. I love you forever.

Chapter 1

Introduction

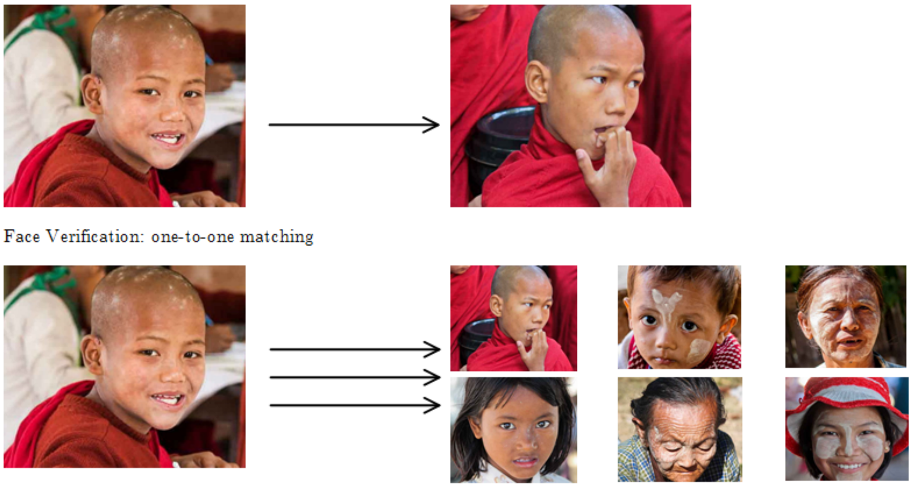
Face recognition, as a daily behavior of human beings, still remains quite a challenge and a major problem when applied to computer science. It associates with many fields like computer vision, biometrics, machine learning and image/video processing. By successfully identifying and verifying individuals from digital images or video clips, the facial recognition system is able to be widely employed to modern society and play a significant role in some specific areas, for example public security, human-computer interaction and video surveillance. Beyond that, researching face recognition which is a spontaneous biological act helps us exploring the mechanism of visual system. Therefore, we focus on this area not only because of its extensive use in the modern society, but also due to its essence which may uncover the secret of how the brain works.

Despite that face recognition is a classic research subject that has been developed since 1960s by Woodrow Bledsoe[2]. It still attracts sizeable number of researchers to contribute to it and shows good momentum of development in the machine learning community. New approaches and improvements of existing ones are continuously put forth. The approaches to solve face recognition problem fall into two main categories. One is mathematical method and another is simulating biological neurocognitive process by utilizing artificial neural network. During the field's first several decades (1960s-1980s), technology had been rapidly developing together with the algorithms, which made it possible for face recognition system to obtain considerable achievements. Since 1990s, with the applying of PCA (principle component analysis) using Eigenface by Kirby and Sirovich[45] as a milestone, mathematical methods for face recognition emerge massively. On the other hand, although the neural networks were strongly connected with face recognition by researchers from the very beginning, they could not gain advantages over classical methods which are more efficient and intuitive. But in 2010s, the neural network methods got reborn with the development of deep learning and general purpose computation. Convolutional Neural Network (CNN) was considered the symbolic approach for face recognition among deep learning methods. In this paper, we are inspired by the CNN framework and integrate its solution into an original mathematical method.

1.1 Face Recognition

1.1.1 Face Verification and Face Identification

In most cases, face recognition as a biometric procedure is referred to both face verification and face identification as shown in Figure 1.1. The decision that which one should be applied or implemented into the system is made mainly based on the purpose.



Face Verification: one-to-one matching

Face Identification: one-to-many matching

Figure 1.1: The difference between identification and verification.

For face verification, we first claim a specified identity and then use the system to confirm whether the face we want to verify matchs this identity. It is a ”if he/she is the person” question. As a result, face verification is a one-to-one feature matching task comparing query image with template image in the database which has established identities. For face identification, on the other hand, we identify the face within all the identities we know (or claim the identity is not known currently). It is a ”do we know this person” question. In order to identify, a one-to-many feature matching task comparing query image with the whole database is employed. For identification, the system is required to find the most similar face or faces with the highest matching scores, or even to report the identity of the query image is unknown if all the matching scores are lower than a specified threshold. In this thesis, we focus mainly on the face verification and plan to extend the method to face identification in the future.

1.1.2 System Structure

For human, brain as a well-connected system is capable of executing the whole recognition process in an instant, but computers lack such ability. So a pipeline which passes results from former process to the next one is constructed in order to deal with the job step by step. It is also helpful to research because algorithms corresponding to each step could be different.

Generally speaking, an integrated face recognition system can be divided into five sections: image/video acquisition, face detection, preprocessing, feature extraction and feature matching. System flow chart is shown in Figure 1.2. First of all, the system should acquire graphics and detect face areas in the images/videos and cut them out from the background. Most of the times, face images/videos which are acquired by camera or other image acquisition devices are not suitable for direct recognition processing because conditions like illumination, angle, face position and gray scale vary. So preprocessing is required to achieve image fixing, alignment and normalization. The next step is extracting features, which contain the most valuable and most stable information to distinguish a person's face from another, from images and organizing them. At last, those features extracted from input face would be matched with enrolled faces'. As we know the identities of enrolled faces, the system would either identify the input face as the closest matching face or claim it as unknown.

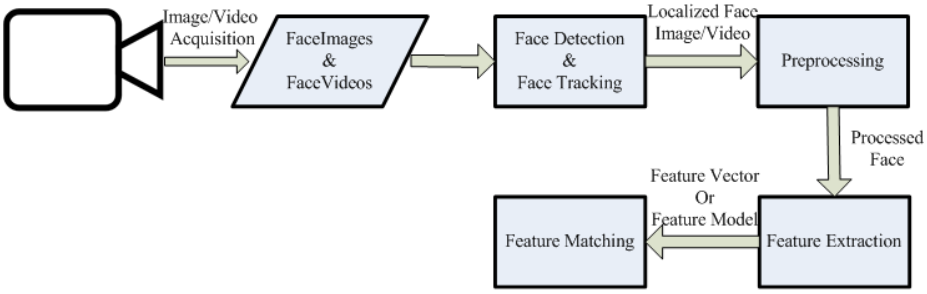


Figure 1.2: A general face recognition system flow chart.

1.2 Challenges of Face Recognition

As we mentioned previously, the development of face recognition shows both spectacular progresses and tough challenges. Algorithms and methods we proposed have already acquired remarkable effect. Therefore, it is safe to say that there are systems that could achieve satisfactory recognition rate under particular circumstances. But if diversity and uncertainty of the environment exceed the capacity of system, the performance would highly depend on how well the system could handle challenges listed below.

1.2.1 Facial Movement

Faces in the daily life, contrary to static images the researchers usually deal with, is changing all the time. So it has always been a challenging task for face recognition methods that the face processing methods applying to static faces could also be used on moving faces. Facial movements are manifold and for different types of movement the system should process in different ways. The two major types of facial movements are rigid movement and non-rigid movement[32]. Rigid movement is also called rigid head motion. It is the moving of head orientation which provides many views of the head instead of static front face (e.g., nodding, looking above and turning around). Non-rigid movement, however, refers to the facial parts' translocation and shape changing which are caused by the movement of skeleton and muscle (e.g., facial expression, speech production and sight changing). Both movements' combination constitutes human face behavior in normal life and delivers more information for communication. However, as face images are two-dimensional graphics transformed from three-dimensional models' projection, the movements would bring uncertainty, complexity and information loss to face recognition.

1.2.2 Illumination Variation

Illumination variation, also known as light direction changing and light condition changing, is one of the major challenges that face recognition system should face. On account of illumination's uneven distribution on faces, the image could vary greatly on luminance, grayscale and shade, partially or globally. This means illumination does not only affect the image in pixel scale, but can also diminish features' information and integrity. Even the problem has been studied for years with enormous progress[39], researches still approve that variation of illumination has no negligible negative effect on face recognition performance[26]. For example, results in FRVT 2006[34] indicate that recognition rate for faces with illumination variation especially uncontrolled illumination variation still has enough room for improvement.

1.2.3 Occlusion

It is very common for human beings to wear accessories or disguises like sunglasses, wig and gauze mask. And human could easily recognize others even under such condition. But for automatic face recognition system, occluded face is a big challenge owing to information loss of the covered part. The issue becomes particularly serious when the recognition algorithm is based on features which are corrupted by the occlusion (such as PCA[3] and LDA[50])[58]. Some methods tend to localize the occluded part then discard features related to that part and others process local features instead of global features so the recognition would not be affected as much by occlusion[55].

1.2.4 Others

There are far more challenges than we could solve once and for all when it comes to uncontrolled face recognition, ageing, scar, makeup, size of the image/video, image/video quality, etc.

1.3 Face Recognition Applications

Although face recognition is a technique facing many challenges and problems, it has now achieved a remarkable amount of reliable solutions for today's applications in personal, commercial or governmental environment. The reasons why face recognition as a biometric is popular are listed below:

- Nonintrusive. Unlike fingerprint, iris or handwriting, face image/video acquisition does not force people to perform specific behavior. The process could be done without being noticed if necessary. This makes it an ideal tool for security and surveillance tasks.

- Compatibility and cheapness. Cameras have high penetration rate not only in public area but also in personal life by being implemented in most laptops and cellphones. So it is easy and economical to build or add a face recognition system. In Machine Readable Travel Documents (MRTD), facial got the highest rate of compatibility among six popular biometric attributes[14].
- Natural. Face recognition is a daily behavior that performed by human all the time, intentionally or unconsciously. This involuntary characteristic gives face recognition convenience when being acquainted by public.

And some applications of face recognition are listed below:

- Security and surveillance. CCTV control system and video surveillance system are widely used in public and private property area with face recognition system involving. Canada Border Services Agency is developing facial recognition system as part of the self-service border clearance kiosks program in major Canadian airports, which will improve security reliability and raise efficiency[28].
- Social activity and entertainment. Face recognition based on cameras is a new and better method of human-computer interaction. Facebook can provide tag suggestion for user photos which is based on facial recognition software[27]. Their face recognition project called DeepFace could reach an outstanding accuracy of 0.9735[49].
- Access control. This includes system login and area entering. Face, as a reliable identify label, has good compatibility with other access control methods like

password or key. It further provides higher reliability and lower risk of key stolen or password leaked. The requirement of recognition accuracy is not so strict thanks to the small number of expected access objects. As an example, Microsoft implements Windows Hello Face Authentication into Windows 10 to make user login safer and faster[29].

1.4 Research Background

As described in the history of face recognition, Convolutional Neural Network(CNN) is the one that bring neural network methods back to the academic view of computer vision and machine learning by showing striking outperformance on ILSVRC-2010 benchmark[18]. From what we learn, deep learning especially CNN has achieved great performance on computer vision and face recognition[33][47][43][49][48]. CNN, which is inspired by receptive field and varied from multilayer perceptron, is a combination of classic ideas[22] like locally connecting, convolutional filter, pooling and several new concepts like dropout[25], overlapping, Rectified Linear Units(ReLu) and GPU computation. Each convolved filter could be viewed as a way of feature extraction. And the convolutional process is selecting corresponding high activate value part from the image. The outstanding capacity of feature extracting utilizing convolution is one of the key factors that helps CNN to achieve great performance of computer vision problems.

In 2010, a face recognition method that based on Pairwise Macropixel(a small pixel block that contains a few pixels) comparison is proposed by Liang Chen[6]. It shows that this pixel block matching approach, even at its very early stage of development, could perform no worse than other mainstream holistic approaches. However, it is irrefutable that comparing macropixel directly is a shallow approach and requires further improvements. By investigating the approach, we find that Macropixels are more like generalized, fixed-size, indiscriminate features for face image. According to this line of thinking, there is resemblance between Pairwise Macropixel Comparison and feature matching. Therefore, considering CNN owns decent feature extracting ability, we wish to explore if CNN's convolutional method could benefit the macropixel approach.

1.5 Research Contributions

There are three major contributions. First, considering the inherent relation between macropixel and basic features in conventional neural network, we propose a face recognition approach based on macropixel approach and inspired by convolutional neural network solution. Two key features are introduced into the original macropixel method: Heavily Overlap and Weight Filter. The former which is inspired by convolutional operation extracts macropixel in an overlapping way. The latter inspired by convolutional kernel trains a filter and uses this filter in the recognition phase. Results from the experiments which are set under the same condition of original macropixel method demonstrate that our proposed approach achieves markedly better recognition rates than the original method. Second, as heavily overlap and weight training requires more computing resources, we also introduce parallel computation into the system using graphics processing unit(GPU) as a demonstration that shows the efficiency potential of the macropixel approach. Third, this research explores the possibility and potential of an original face recognition approach. It also proves that the ideas of deep learning framework are possible to benefit traditional face recognition method, which motivates us to further study the relationship of these two kinds of approaches.

1.6 Outline of the Thesis

The structure of this thesis is organized as follows:

In chapter 2 previous related works in the field of face recognition are reviewed. It contains a brief description of the system, followed by several current recognition approaches.

In chapter 3, researches that related to our work are discussed. The first part is focused on the Macropixel comparison approach and analysing the characteristics of it. The second one mainly provides the introduction of convolutional neural network together with state of the art progresses of it.

In chapter 4, two approaches are proposed. The first is to import overlapping with different degrees into the comparison and the second approach is to further add weight filter. Different tentative methods for each approach are discussed.

In chapter 5, The procedure of our experimental setup as well as the exploration of graphic processing is described in detail at the beginning. Then the experiment results of proposed approaches are presented step by step. We first report the performance from each approach separately, and then combine them together in order to get an outperformance. This chapter also provides efficiency test and analysis afterwards.

In chapter 6, thesis and research is concluded and future work is discussed.

Chapter 2

Literature Review

In this chapter, we will first go through the processes of face recognition in general. As a complicated multistage system, it requires us to discuss some stages a little further to get a more complete picture. And then we will focus on the most essential stage of the system, which is recognition stage (including feature extracting, feature matching and classification). Several recognition algorithms that utilize different methods to comprehend and paraphrase characteristics of faces are introduced.

2.1 The Face Recognition System

We have already provided a brief flow diagram for face recognition system in Figure 2.1. As shown in the flow diagram, the recognition system contains image/video acquisition, face detection, preprocessing, feature extraction and feature matching. If we go a step further and take training process into consideration, pipeline of the system is as depicted in Figure 2.1. Both training and recognizing should go through the same processes until the last step. The resulting features of training are stored in the database in order to be matched with features extracted by images waiting to be recognized.

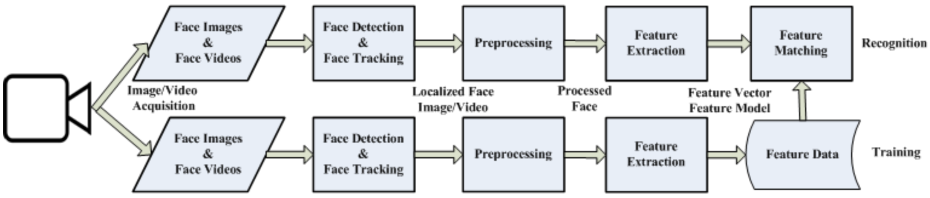

Face Images & Face Videos
Face Detection & Face Tracking
Preprocessing
Feature Extraction
Feature Matching
Recognition
Image/Video Acquisition
Localized Face Image/Video
Processed Face
Feature Vector Feature Model
Face Images & Face Videos
Face Detection & Face Tracking
Preprocessing
Feature Extraction
Feature Data
Training

Figure 2.1: A general face recognition system flow chart with training process.

2.2 Face Detection

The primary task of face recognition, after it has acquired images/videos, is to detect face's location. Face detection is a fundamental technology for computer vision as it is not only necessary for face recognition, but also indispensable for auto-focus tracking, photo retouching, face modeling, expression recognizing and many other facial analysis processes. According to M. H. Yang, D. J. Kriegman, and N. Ahuja's definition, "the objective of face detection is to identify if there are faces in the given image and, if present, return face location and extend of each face"[56]. This definition clearly indicates that the inputs of face detection are images and the outputs are coordinates and images of face. As with face recognition, face detection could also be disturbed by face's position, scale, illumination, expression, etc.

From what we learn from previous surveys for face detection, researchers in this field have achieved proposing more than 100 approaches, including some outstanding ones that are widely used in people's lives today[56][57]. As a famous classifier, SVM (Support Vector Machine) is used on face detection in Osuna and Heisele's researches[31][13]. The former implements SVM into face detection together with a decomposition algorithm to overcome SVM's low efficiency when dealing with large data set and the latter comes up with a two-level SVM classifier that has better result for face rotation problem. Then Romdhani and Ratsch speed up the calculation of SVM based algorithm using reduced set vectors[36][37]. Other than that, a new Bayesian Discriminating Features method proposed by Liu uses Bayesian classifier to distinguish face class modeled as multivariate normal distribution from no-face class[24]. That algorithm only models no-face class that is closest to face classes and excludes most of the no-face objects. Paul Viola and Michael Jones proposed a detection algorithm using Integral Image as image representation and AdaBoost as learning method[51]. Integral Image approach is constructing an image representation that allows fast selection for partial rectangular area, which makes it easier to compute Haar-base features. This detection approach had much imrpoved operational efficiency comparing to previous ones and it had become widely used in real life afterwards. Neural Network has also been applied on this area, from Rowley's classic Neural Network-based filter[38] to Garcia's convolutional neural architecture[11].

2.3 Preprocessing

As mentioned earlier about challenges face recognition may suffer from, some of these challenges could be very difficult, or almost impossible to overcome if we leave them to feature extraction or matching phase. So preprocessing (face normalization, noise elimination, lighting variation correction, etc.) plays a significant role in face recognition to deal with those kinds of problems.

Among all the preprocessing approaches, traditional image enhancement methods have the advantages of not requiring prior knowledge and processing directly and quickly. So they are commonly used by many face recognition systems. Approaches based on modifying Histogram (for example Histogram Equalization) are popular dealing with illumination variation[41]. Histogram equalization redistributes the histogram into a uniform one in order to enhance local contrast of the image to readjust image illumination condition[35][8]. Histogram Specification/Matching transfers the histogram into a predefined form that is proved to have a better illumination condition. Logarithmic Transformation stretches low grey part of the histogram and condenses high grey part to get better visual adaptation. Aside from classic histogram methods, Gamma Intensity Correction was introduced by Shan to correct the overall brightness of the face images to a pre-defined ”canonical” face image[44].

Face Geometric Normalization is another kind of important technique in face preprocessing. Its purpose is normalizing face images to get fixed eye/nose/mouth position, uniform face orientation, same image size, etc. The normalization is usually carried out by selecting a facial feature and modifying image based on it. The most commonly used feature is eyes and the methods to modify the image are clipping, rotation or scaling[5].

2.4 Feature Extraction

Feature extraction is aimed at acquiring the most representative information that could be used to distinguish one face from another from a facial image. In the study of face recognition, stage of feature extraction has gotten extensive attention because of its most important role in the whole system. Therefore, researchers keep exploring possibilities that could improve the performance of this stage including choosing a feasible feature for extracting and employing a proper way to extract and describe the feature.

The methods of extraction have been divided into two categories. One is holistic representation and another is local feature[42]. Holistic representation is focusing on the whole image which could be treated as a high dimensional feature space. Feature vectors from the space are the features extracted. Because holistic features represent the whole face, there are feature vectors that hold little effective information for face recognition which might put a drag on system efficiency as well as recognition accuracy. Local features are facial characteristics and their location information including lip, eyes, nose and their relative positions.

Here we will introduce some of the most representative extraction methods for both categories.

2.4.1 Eigenfaces and Principal Component Analysis

The original developer of Eigenfaces method is Sirovich and Kirby who found it effective as a face feature representation[45]. This holistic approach is one of the most widely used algorithms in face recognition. The key idea behind Eigenfaces is to project face images in pixel space into a subspace which has lower dimension but still holds significant characteristics of faces. The key method of space projection is Principal Component Analysis (PCA). We first get eigenvectors from the covariance matrix of facial images by eigenvalues decomposing. Then we use eigenvectors (i.e. eigenfaces) with the largest eigenvalues as the features to represent characteristics or variation of faces. Each image could be projected into the eigenvector space, in other words, by using the linear combination of these basis eigenvectors, we can get an expression of each image and use that expression for comparison and recognition. By reducing the dimensionality of input data, PCA approach removes redundant information from features, gets higher feature stability and increases the computation efficiency which as a result improves recognition performance.

Here is the example of steps of PCA algorithm for face recognition. Giving a set of N face images $S = \{I_1, I_2, I_3, ..., I_N\}$, step one is to calculate the mean face image $\bar{I}$ of all faces. In step two the covariance matrix A is calculated:

$$A = \frac{1}{N} \sum_{1}^{N} (I_i - \bar{I})(I_i - \bar{I})^\top \tag{2.1}$$

The next step is to get the eigenvectors and eigenvalues of covariance matrix:

$$AU_i = \lambda_i U_i \tag{2.2}$$

where U_i is eigenvector of the corresponding eigenvalue λ_i. By sorting eigenvalues in the order of descending, the first N eigenvalues and their corresponding eigenvectors $U = \{U_1, U_2, ...U_N\}$ are selected. The last step is to project input images on eigenvectors' direction:

$$B_i = U(I_i - \bar{I}) \tag{2.3}$$

where B_i is the eigenvector representative of input image I_i. The projected input images will then be used for recognition by comparing Euclidean distance or using classifier.

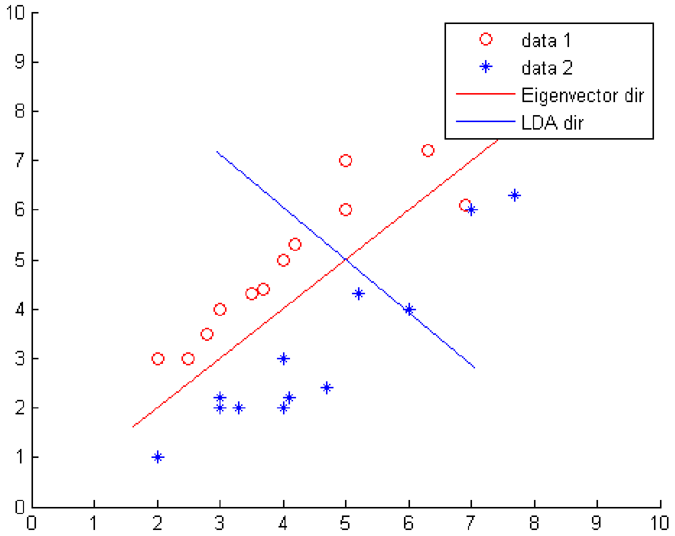


Figure 2.2: PCA focuses on variation within-class while LDA finds direction that maximizes variation between-class.

2.4.2 Fisherface and Linear Discriminant Analysis

Fisherface, which is firstly developed by Robert Fisher in 1936, is also a traditional holistic method for face recognition[3]. It is based on the theory of Linear Discriminant Analysis (LDA) and shares some similarities with Eigenface. They both project the input data on a low dimensional space in order to extract features. But differs from PCA which focuses on variation of features, LDA approach's purpose is to find vectors that maximize discrimination among classes instead of feature differences within class. The difference between PCA and LDA has been shown in Figure 2.2. While the direction got by PCA has both classes mixed, the LDA finds the direction that distinguishes data from two classes at the extreme.

We can represent the within-classes scatter matrix of input data as:

$$S_w = \sum_{i=1}^{C} S_i \tag{2.4}$$

And S_i can be defined as:

$$S_i = \sum_{x \epsilon C_i} (x - m_i)(x - m_i)^\top \qquad (i = 1, 2, ..., C) \tag{2.5}$$

where m_i is the mean of class C_i, Pi is the number of samples for the class. In order to make classes depart from each other, in other words, get bigger between-class scattering and gather together data in the same class, in other words, get smaller within-class scattering, we can get the Fisher criterion function below:

$$J\left(W_o\right) = \frac{W^\top S_b W}{W^\top S_w W} \tag{2.6}$$

When maximizing J, the corresponding eigenvectors is what we are looking for.

2.4.3 Independent Component Analysis

PCA is based on the second-order relationships of input data but the high-order relationships are ignored. However, in many situations, the discriminative information for face recognition lies in the high-order relationships of images. Therefore, Independent Component Analysis (ICA), which is a generalization of PCA and makes full use of high-order relationships information, is brought into view of face recognition research[17]. The idea behind ICA is to decompose the input data into a group of independent components through linear transformation. Comparing to PCA, ICA's basis vectors can better represent localizes features and it is able to apply to high-order decorrelation. These advantages make it has better performance than PCA in many cases.

2.4.4 Gabor Wavelets and Elastic Bunch Graph Matching

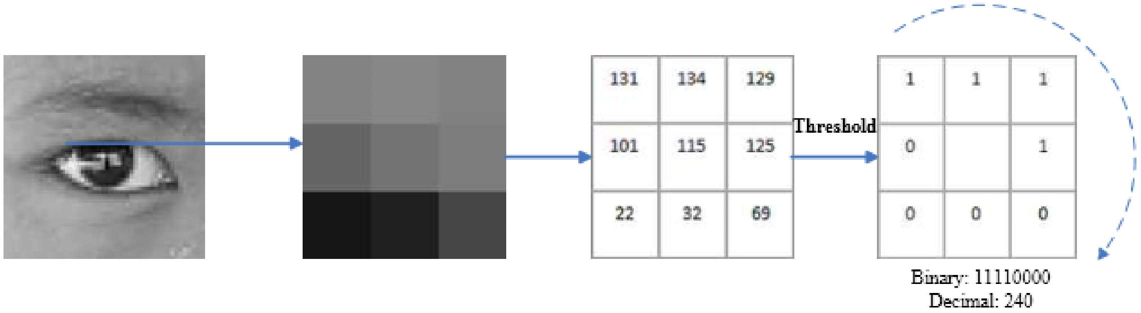
Dynamic Link Architecture(DLA), which was proposed in 1981, was meant to fix a deficiency of neural network that the way to express relations among different types of active neurons is unclear[52] and Lades first implemented it into face recognition system in 1993[19]. The core idea of DLA was inspired by the plasticity of synapse: neurons could be formed into graphs which contain a group of connected nodes, and both nodes and their connections vary rapidly in the short term or slowly in the long term. In order to achieve this, two kinds of weights, T_{ij} and J_{ij} (where I and j are connecting neurons), are introduced. J_{ij} changes rapidly with the constrain of T_{ij} ($0 \leq J_{ij} \leq T_{ij}$) corresponding to short-term memory meanwhile T_{ij} changes slowly in a long time scale representing permanent memory and weight of conventional neural network.

For DLA method, Gabor-based wavelet is chosen as the local feature representation. Gabor wavelet is a kind of complex wavelet modulated by Gaussian function. Researchers found that it is similar with the response comes from cells in biological vision system[9]. And the wavelet, as a feature representation, has strong robustness to rotation, scaling and distortion as well as good adaptability to illumination changes. So it is widely used in the field of computer vision. The DLA approach first chooses several local feature pixels or regions like eyes, nose and shape of jaw then extracts them with Gabor wavelets as Gabor ”jet”. After that, a bunch graph with the shape of rectangle is created containing nodes correspond to jets. Comparing jets’ similarity of two faces is the way of recognition.

Elastic Bunch Graph Matching (EBGM) was then developed based on DLA by Wiskott[54]. Instead of containing only one jet for each node, the node now attaches a bunch of jets extracted from different face images in order to locate corresponding features for different faces. EBGM approach got better result and it soon became one of the most popular feature extraction approaches among local feature based methods. However, both DLA and EBGM have the weakness of large calculation quantity.

2.4.5 Local binary Pattern

Local Binary Pattern (LBP) is another feature extraction method proposed by Ojala in 1996[30] and used for face recognition by Ahonen in 2004[1]. This approach uses LBP operator to build LBP feature map for comparing and classifying. The original LBP operator is composed of a round of eight neighbor pixels and value of pixel locates in the center is the threshold. Any surrounding pixel with a higher gray value (or equal) than center one's will be assigned number 1 and others get number 0. The LBP code is represented by the binary number sequence after processing which contains basic feature information inside. The principle of LBP shows in Figure 2.3.



131	134	129
101	115	125
22	32	69

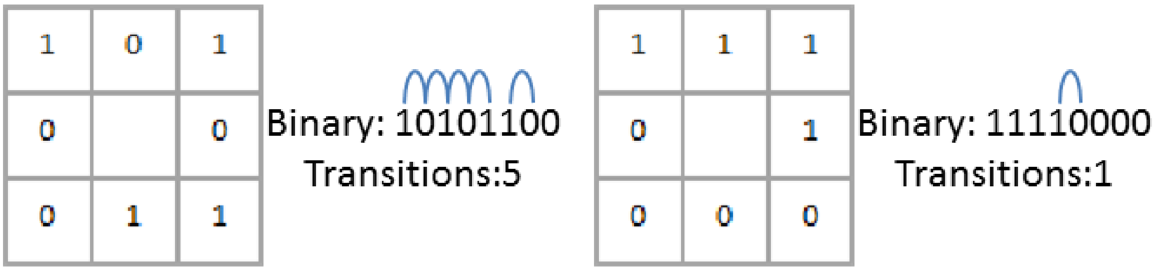
1	1	1
0		1
0	0	0

Binary: 11110000
Decimal: 240

Figure 2.3: The process of LBP feature extraction. The decimal value represents the texture information.

Improvement for LBP was then introduced including using different size of operators, different number of operator's neighbor pixels and uniform pattern. A uniform LBP contains pattern with at most two bitwise transitions from 0 to 1 or vice versa

(Figure 2.4). By using uniform LBP, both computing resource and memory are spared. And only important basic features like edge, corner and spot are selected.



1	0	1
0		0
0	1	1

1	1	1
0		1
0	0	0

Figure 2.4: The left one which has 5 transitions and in only assigned to a single label. The right one with 1 transition(transitions ≤ 2) is an uniform LBP.

2.5 Classification

After getting features extracted from face images, we could either use it for matching directly or build a classifier for face classification. There are a large number of classifier been proposed for face recognition, and we choose some of the representative methods to introduce.

2.5.1 Support Vector Machine

Generally speaking, Support Vector Machine (SVM) is a binary classification model that separates two classes in the most perfect way. It selects a boundary hyperplane to maximize the margin (the minimum distance between hyperplane and features on both sides). In order to achieve this, two parallel hyperplanes that push against nearest features are introduced. The distance between boundary hyperplane and either one of the parallel hyperplanes is the margin. Therefore, the larger the value of margin is, the better is the separability of two classes.

2.5.2 Neural Network

As the biological principle being applied on the bionic theory and computer science technology, artificial neural network is born and has been deeply explored and widely used in the field of artificial intelligence. Benefits from its architecture, classifiers based on neural network are naturally applied to face recognition. Artificial neurons are the basic components connected with each other using weights to control the passing information. A lot of architectures for neural network have been proposed including feedforward neural network, back propagation, multilayer perceptron and deep neural network. Convolutional neural network is currently the focus in face recognition researches and we will discuss it further more in the next chapter.

Chapter 3

Previous Work

In this chapter, we will introduce previous works that founded and inspired our research. The foundational approach utilizing small pixel blocks as features for recognition named ”Pairwise Macropixel Comparison” is introduced at first. Its characteristics are then discussed. The next section will focus on discussing Convolutional Neural Network (CNN) which gives us ideas about how to improve the Macropixel approach. The general architecture of CNN is introduced together with some state-of-the-art research progresses.

3.1 Pairwise Macropixel Comparison

In 2D face recognition system, pixel is the fundamental information carrier for human faces, whether they are represented by images or videos. However it is self-evident that single pixel is not able to provide effective information for recognition process.

But on the other hand, images with a few pixels might be able to represent some information for face features. As a result, urged by the question that whether small pixel block could be used for face recognition by comparing holistically, the idea of Macropixel is proposed. Research objective of the method is to explore if it can get no worse result than other holistic methods, which also implies that, since there is no subspace or dimension reduction applied, this currently developed technique may not be necessary for face recognition process[6].

3.1.1 Definition

The Macropixel approach is built based on several key concepts, which not only explain terminologies but also provide a complete blueprint for the designed process. These definitions are explained below:

Preprocessing: Face geometric normalization is employed previously in order to get face images with fixed orientation, size and position using eyes as center alignment. It is assumed that each image after geometric normalization has a size of $N \times M$. As conventional image enhancement methods support direct processing without changing data structure mentioned in 2.3, they are used for Macropixel for the sake of comparing with other approaches utilizing the same pre-processing techniques. As default, vector normalization is utilized on macropixel to eliminate the illumination variation.

Macropixel and Shifting: Macropixel is a group of pixels segmented from the targeting image in the shape of square. The size of macropixel is set to $K \times K$ (the default number of K is 4). Macropixel is transferred into 1 dimensional vector $V = [v_1, v_2, ..., v_x]$ where $x = K^2$ for distance calculation. For the purpose of aligning corresponding macropixel features between face images, there is a margin area with size of s pixels (the default number of s is 2) at each side of the image. The macropixel and shifting is shown in Figure 3.1.

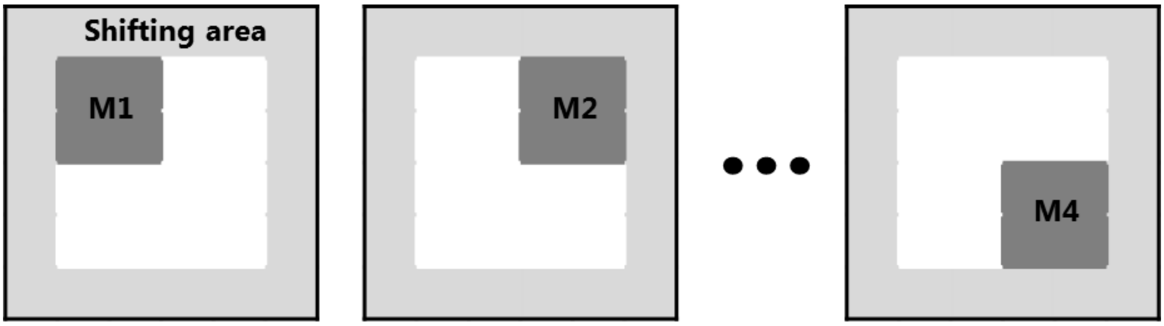


Figure 3.1: Assuming image with size 12 × 12, the 2 pixels margin area are for shifting purpose, and we segment 4 macropixels with size 4 × 4 from this image.

Distance: Measuring similarities is a major task for data analysis, which also applies to the comparison of Macropixels. The Euclidean distance between two macropixels is chosen as the method of similarity measurement[10]. Euclidean distance $D(V_1, V_2)$ between two macropixel vectors $V_1 = [v_{11}, v_{12}, ..., v_{1x}]$ and $V_2 = [v_{21}, v_{22}, ..., v_{2x}]$ is defined as:

$$D(V_1, V_2) = \sqrt{(v_{11} - v_{21})^2 + (v_{12} - v_{22})^2 + ... + (v_{1x} - v_{2x})^2} \tag{3.1}$$

$$D(V_1, V_2) = \sqrt{\sum_{i=1}^{x} (v_{1i} - v_{2i})^2} \tag{3.2}$$

With a shorter distance, the degree of similarity is higher.

Corresponding macropixel: Assuming there is a macropixel M whose left top corner's coordinate is (x_0, y_0) in image A, the eligible macropixels in image B are defined as ones with left top corner's coordinates ranging from $(x_0 - s, y_0 - s)$ to $(x_0 + s, y_0 + s)$. After calculating distances between M and the eligible macropixels, the one with the shortest distance is selected as the corresponding macropixel of M.

The candidates and corresponding object are shown in Figure 3.2.

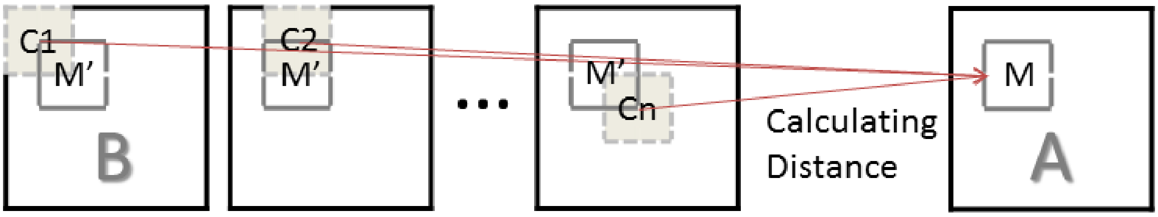


Figure 3.2: Calculating distances between macropixel M in image A and candidates $Ci(i = 1, 2, ..., n)$ in image B. The corresponding object of M is among Ci and has the shortest distance.

3.1.2 Recognition Process

The recognition pipeline of Macropixel approach is shown as follows:

a Query image A is the input face image waiting for identification, and the system stores images with different identities as templates. All images have the size of $N \times M$.

b A counter is set for each identity in the template images.

c Assuming $N - 2s$ and $M - 2s$ are dividable by K, image A is segmented into a number of R macropixels which have a size of $K \times K$ leaving margin area with the size of s pixels for shifting. $R = ((N - 2s)/K) * ((M - 2s)/K)$.

d Sequentially picking one macropixel P in A, the system finds corresponding macropixels from all template images by calculating and comparing the distances between P and eligible macropixels.

e After comparing distances between P and all the corresponding macropixels, the one with the shortest distance is selected together with its identity T.

f The number of identity T is increased by 1 in the counter.

g System repeats processes of d, e, f until all macropixels in Image A have been traversed.

h Query image A is identified with the Identity which has the largest value in counter.

3.2 Convolutional Neural Network

We have briefly introduced deep learning and CNN in chapter 1. As an improved architecture of neural network, deep learning is able to illustrate complex concept or object by comprehending more basic fact through layers, which is the state-of-the-art achievement in the field of constructing or simulating artificial intelligence. Among all deep learning methods, CNN is one of the major framework of deep learning. Benefiting from its inherent relationship with visual cortex, it has achieved great progress in computer vision. CNN is firstly motivated by Hubel and Wiesel's biological study of animal visual cortex[15]. They not only discovered the cerebral hierarchical mechanism of visual neurons. But also, the concept of receptive field, which means a sensory region whose stimulus could make particular neuron responses, was introduced into the research. Then based on the Multilayer perceptron(MLP) architecture and Backpropagation(BP) algorithm originally proposed by Werbos[53] and carried forward by Rumelhart [40] , Lecun proposed the convolutional neural network utilizing gradient descent backpropagation algorithm[21][22] which was then promoted into face recognition[20].

3.2.1 The general framework

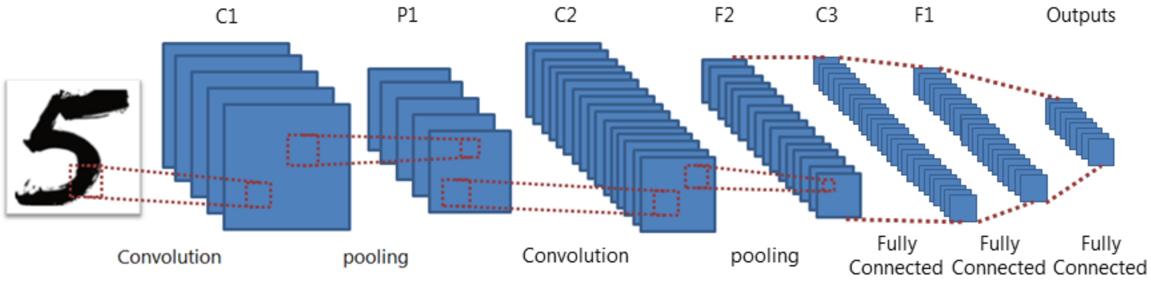


Figure 3.3: LeNet-5 is a typical Convolutional Neural Network framework. C1, C2 and C3 are convolutional layers. P1 and P2 are pooling layers. F1 is fully-connected layer.

Figure 3.3 shows the typical architecture of LeNet-5[22], which is a classic CNN model used for digital recognition. The major frame of the first part is a series of feature extracting and feature mapping procedures operating in convolutional layer and subsampling layer respectively. The system gradually uses a number of detailed feature maps (like edges or blobs) as inputs coming from the previous layer and extracts local features from them, which is then used for constituting more abstract feature maps. In the second part, the features are imported into fully-connected layers which are equivalent to traditional MLP for the purpose of classification. Each stage of the network is further explained in detail below:

Convolutional Layer: As a variant of feedforward neural network, CNN distinguishes itself because of two properties: sparse connectivity and shared weights. And these two ideas are implemented in the convolutional layer via convolutional kernel operation (see equation 3.3). The kernel is a linear filter with a set of weights and it only connects to one sub-region of the input feature map each time instead of fully connecting. By executing the computation repeatedly through the input image with the same kernel which equates to convolution of the input, a corresponding output feature map is produced. Hence, one feature map is generated by only one kernel, which reduces the computational burden. Generally, the output consists of multiple feature maps to have a comprehensive representation of the input.

Assuming a 2D input feature map in layer l is h^l (with size $M^l \times N^l$) and each sub-region is x_{ij}^l with a stride of a, the output feature map is h^{l+1} and each sub-region is x_{ij}^{l+1}, weighted filter is W (with size $K \times L$), the bias is b, the convolutional operation is shown below:

$$x_{ij}^{l+1} = A(W * x_{ij}^l + b) \tag{3.3}$$

Where $i \leq M^{l+1} = (M - K + a)/a$, $j \leq N^{l+1} = (N - L + a)/a$ and size of feature map h^{l+1} is $M^{l+1} \times N^{l+1}$. $A(x)$ is activation function that introduce non-linearity into the network, for example $A(x) = max(0, x)$ (Rectified Linear Unit function [12]) or$A(x) = tanh(x)$.

Pooling Layer: Pooling layer, which is also called subsampling layer, is meant to compress the feature map by merging sub-regions into one value. There are many methods for pooling like average-pooling or sum-pooling [4] but the typical one is max-pooling that exports the maximum value from the sub-regions. The purpose of pooling is to eliminate unnecessary and redundant information from the feature map and preserve important ones. This leads to better robustness over position and lighting condition as well as reducing computation in the next layer.

Fully Connection Layer: While the lower layers are several convolution and pooling layers, the upper layers have convolutional and traditional MLP architecture where neurons connect to all units in the previous layer. They play roles of classifiers mapping extracted features to the most correlated class.

Training Process: CNN's training process is based on BP algorithm. BP is a method employed to pass cost function back through network by using gradient descent algorithm and update weights in order to minimize the cost. The system would propagate forward through the network in order to get the outputs which are used to calculate cost with the labeled correct values. Then it propagates backward calculating cost of each neuron's output with gradient. Ultimately, weights are adjusted based on the costs. But unlike in traditional multilayer neural network, BP method in CNN network requires simulated operations restoring the status of hidden layers before pooling or convolution in order for propagating backward.

3.2.2 Improvement of CNN

All Convolutional Net: In 2015, Springenberg proposed a CNN architecture replacing pooling layers with stride convolutional layers[46]. Since pooling layers' major function is to reducing the spatially dimension of feature maps, he came up with an idea to build an all convolutional net serving the same purpose as the convolution-pooling structure. The result shows that all convolutional net could get better result sometimes and pooling layers are not necessary for CNN.

Global Average Pooling (GAP): A network structure Network In Network (NIN) is proposed employing global averaging pooling layer instead of fully-connected layer[23]. In the approach, each feature map is a confidence map representing a specific category by using the average value of the feature map as the label which is called the confidence value. It significantly reduces the scale of network and alleviates overfitting.

Chapter 4

Proposed Algorithms

Although macropixel method introduced in chapter 3 has been approved to be competitive among mainstream holistic approaches, it is still in its very primitive stage and requires development. So encouraged by the promising result comes from Macropixel Comparison Approach, we explored the improvable possibilities of it and proposed a new convolutional macropixel approach for face recognition.

4.1 Convolutional Macropixel Comparison Approach

In the research of CNN, convolutional layer has been found to be extremely effective in extracting features from face images. Inspired by structure of the layer, two key concepts are introduced into the original approach of macropixel: deeply overlapped comparison and weighted filter counter. Both of them serve the purpose of obtaining richer feature information from macropixels. The experiments and results in chapter

5 show that even though utilizing one of them could not guarantee a better result comparing to the original one, combining them together would significantly improve the recognition rate. Figure 4.1 shows the pipeline of our approach.

4.2 Overlapping: A Convolutional Way

Overlap representation of features is proved to be effective for improving the performance of face recognition methods based on low-level features[7]. Biological evidence also indicates that the animal vision system utilizes overlapping receptive fields, which is then developed into the convolutional operation in CNN framework[16]. So in our work, an overlapping extractor is applied to the method of macropixel comparison.

A standard macropixel approach performs extracting with non-overlapping which was shown before in 3.1. We keep the extracting but add overlap operation onto it. Instead of skipping macropixel's size K pixels to the next macropixel, the system skips less number of pixels to the next macropixel which results in overlapping extraction. Parameters $p, q \in [1, K)$ are set to restrain the stride of extractor, where K is the length of macropixels. For the extreme configuration, p and q are set to 1 in order to perform deep overlap for macropixel extraction and comparison.

As shown in Figure 4.2, for image A with size $N \times M$ and shifting number s, the macropixels $(N \times M)$ extractor is convoluting through the whole image. In original method, the total number of macropixels for A is $R = ((N-2s)/K) * ((M-2s)/K)$. After the implantation of overlapping extractor, the amount of macropixels is:

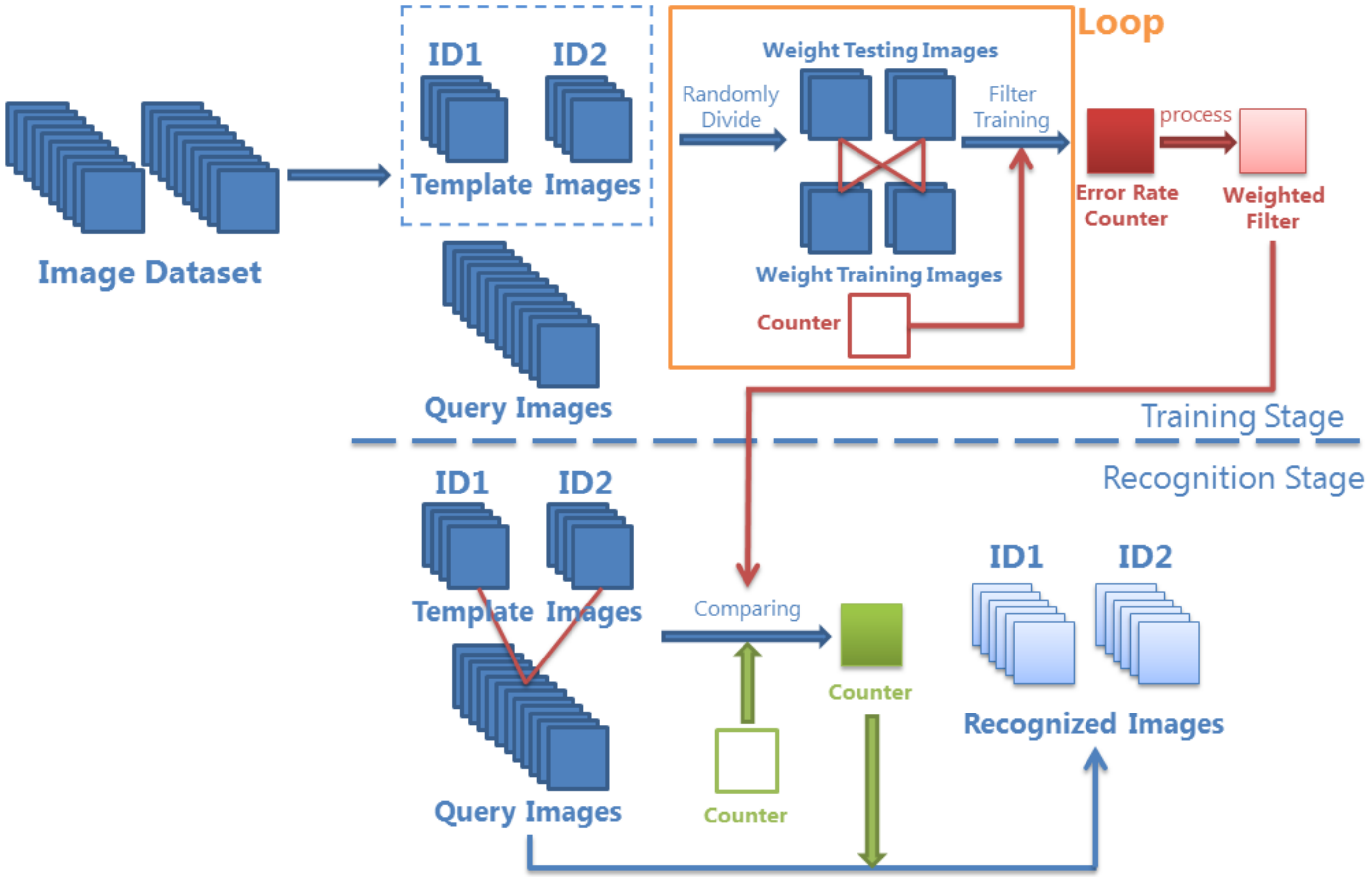


Figure 4.1: The Whole Framework is shown above. The system is divided into two stages: training stage and recognition stage. The training stage produces weighted filter which is used on original recognition. One thing to note here is that all macropixel comparison operations are employing deep overlap.

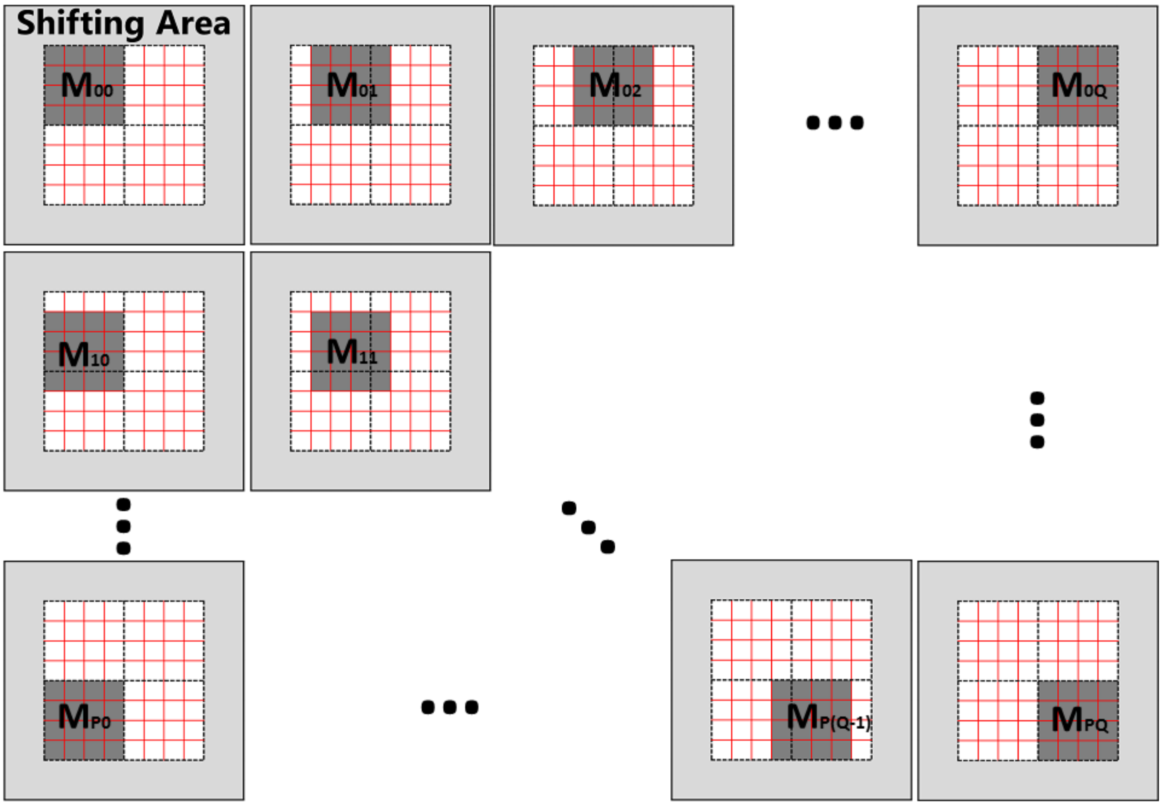


Figure 4.2: For image with size 10 × 10(the effective area is 8 × 8 removing 2 shifting space) and macropixel with size 4 × 4, deep overlap will extract 5 × 5 macropixels out. In this example, $P = Q = 5$.

$$R' = ((N - 2s - k) \div p + 1) * ((M - 2s - k) \div q + 1) \tag{4.1}$$

Overlapping of macropixels provides complete low-level feature information for comparing. But it also brings more non-significant and interfering features into the system, which might leads to higher verification error rate. In chapter 5, it is shown that macropixel method with overlapping extractor solely accomplishes fluctuated performances comparing to the original one. Accordingly, the weighted filter is employed to filter macropixels that do not benefit recognition.

4.3 Weighted Filter Counter

As mentioned in 1.4, macropixel can be treated as discriminate features of faces which means that all features have equal importance. However, in face recognition, there are features that contribute a lot in distinguishing one face from another and features that provide less useful information. Therefore, a weighted filter is introduced filtrating macropixels with higher recognition error rate.

In order to implement weighted filter, the system is composed of two stages: training stage and recognition stage. The training stage uses template images to calculate the weighted filter based on the error rate of each macropixel's location. The training process would repeat several times in order to get a stable weighted filter. The filter is then used on the counter in the recognition stage. Recognition stage is almost the same as the original macropixel approach except the counter is connected to the weighted filter.

4.3.1 Training Stage

Weight Training Operation: Template images are randomly divide into two sets: weight training set and weight testing set. Each set contains a certain number of images for each identity. The weight training operation is executed using these two sets the same way as the original macropixel comparison operation.

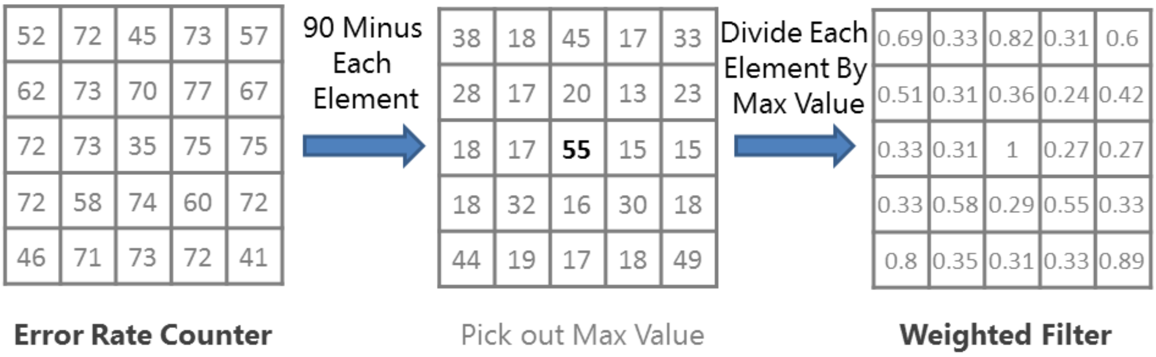
Error Rate Counter: The purpose of training stage is to figure out the error rate on each macropixel's location. With a higher error rate, macropixel at that location is more unreliable in face recognition. In order to do so, we employ a new counter which does not count the value of the identity with shortest distance in weight training operation, but counts if macropixel with specific coordinate is recognized as the wrong identity. The counter C_e is built in the form of matrix whose size is the same as the amount of macropixels extracted from one image(4.1). Each element $c_{ij}(0 \leq i \leq P, 0 \leq j \leq Q)$ in the matrix counter is corresponding to the coordinate of the macropixel M_{ij} in image.

Weighted Filter: The weighted Filter is produced on the basis of error rate counter after training process has finished. First all elements in counter are used to subtract the total number of weight testing images O_{test}. Then we find the maximum value c_{max} among all elements in C_e. Then all elements are divided by c_{max} and we can get the weighted filter O'_e:

$$c'_{ij} = (O_t - c_{ij}) \div max(C_e) \tag{4.2}$$

The weighted filter is shown in Figure 4.3.

The pipeline of training stage is shown as follow (Figure 4.4):



Error Rate Counter

52	72	45	73	57
62	73	70	77	67
72	73	35	75	75
72	58	74	60	72
46	71	73	72	41

Pick out Max Value

38	18	45	17	33
28	17	20	13	23
18	17	**55**	15	15
18	32	16	30	18
44	19	17	18	49

Weighted Filter

0.69	0.33	0.82	0.31	0.6
0.51	0.31	0.36	0.24	0.42
0.33	0.31	1	0.27	0.27
0.33	0.58	0.29	0.55	0.33
0.8	0.35	0.31	0.33	0.89

Figure 4.3: Here is how to transfer an Error Rate Counter to Weighted Filter. The number of weight testing images O_test is 90 and we make 90 minus each value in the counter to get an intermediate matrix. After pick out the max value 55 from the matrix, we divide each element by 55 to produce weight filter.

a All the template images T (with a number of O) are randomly divided into training images for weight $O_{train} = \{a_{t1}, a_{t2}, ..., a_{tn}\}$ and testing images for weight $O_{test} = \{b_{t1}, b_{t2}, ..., b_{tm}\}$ where $O = m + n$.

b Set a new error rate counter $C_{en}(n = 1, 2, ..., X)$ where X is the repeat times of training.

c O_{train} and O_{test} are going through the original macropixel comparison process for the first macropixel.

d For any image in O_{test} that is identified as a wrong person, error rate counter's element value for the corresponding macropixel would increase by 1.

e System repeatedly executing step b and c until the last macropixel.

f The weighted filter C'_{en} is generated using counter with fomula 4.2.

g System repeatedly go through step a to step f X times. By adding all weighted filter together and getting the average value, the stabilized weighted filter is produced:

$$C_{final} = \sum_{1}^{X} C'_{en} \div X \tag{4.3}$$

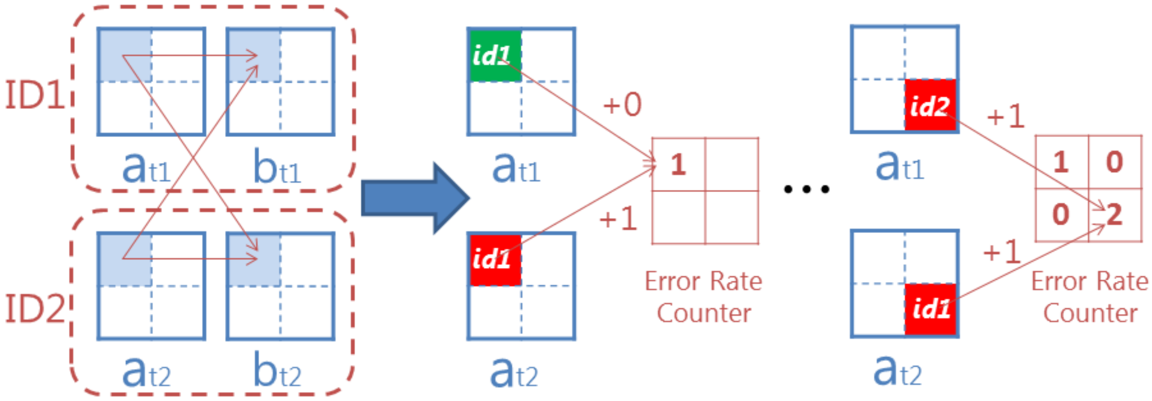


Figure 4.4: By comparing between training images a_{t1} and a_{t2}, the system identify the first pixel of a_{t1} as ID1 which is a success so it adds 0 to the counter. For image a_{t2}, the identification operation fails so system adds 1 to the counter. The total value for counter in the first location is $0 + 1 = 1$.

4.3.2 Recognition Stage

We have explained the process of original macropixel approach in 3.1.2. There is not much difference in the new recognition stage except step f. In step f for the new method, we pick value of weighted filter's element corresponding to the macropixel the system is executing, and add that value to the counter instead of 1.

4.3.3 Independent Weight Training

Although Weighted Filter is an effective tool that improves performance of Macropixel comparison, its operation requires more computational resources comparing to the original method. To solve this problem, we propose independent training approach in which the training stage is operated individually to get pre-trained Weighted Filter and use it on recognition stage separately. The idea is to train a filter using template images and reuse it on other faces' recognition. The independent training operator is defined as:

$$I_{DB}{}_{wt}^{nt} \tag{4.4}$$

where DB is the data set used for training, nt is the number of template images and wt is the number of weight training images.

From the experiment results in Chapter 5, it is indicated that the pre-trained weight filter produced by one database is of generality and could increase the recognition rate on different databases.

Chapter 5

Experiments

In this chapter, the experiment's scheme is introduced at first. We tested our framework on three databases following the original paper[6] including the UIUC version of Yale face database, the ORL Database of Faces, and The CMU Multi-PIE Face Database in order to measure the improvement of system intuitively. To explore the characteristics of our new algorithm, the performances of system on Yale database with a wide range of different situations are studied. And then we move on to ORL and PIE database which have larger data scale.

5.1 Face Dataset

For the purpose of studying performance of proposed approach comparing to the original one, we continue to use UIUC version of Yale, ORL and PIE Database. The databases contain two size versions (pixel size of 32×32 and 64×64) and have been

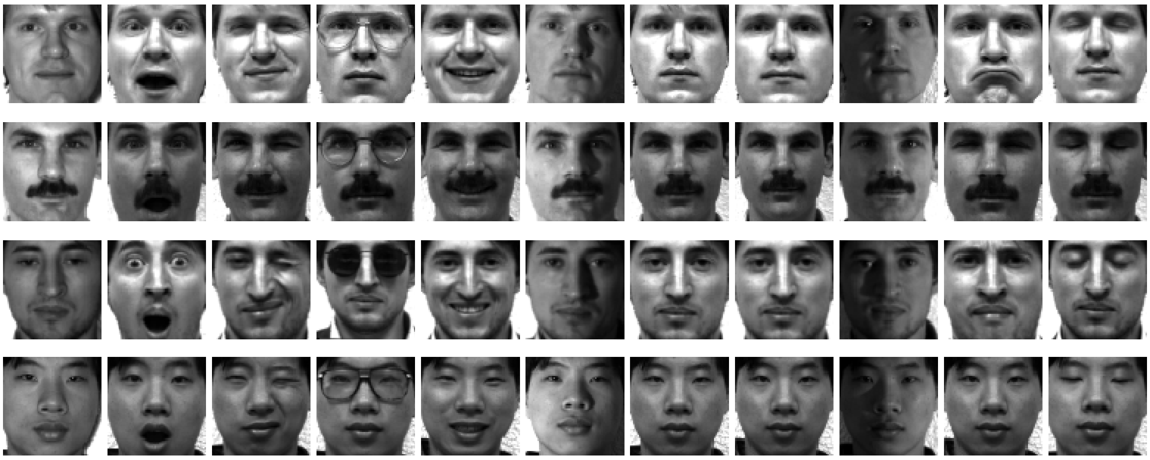


Figure 5.1: Face images of 4 sample subjects from Yale Database. Each subject contains 11 images with different illumination conditions and expressions.

geometric normalized.

Yale Face Database: Yale is a classic database with 15 subjects and 11 images for each subject (see Figure 5.1). These 165 grayscale images are with different facial expressions, various illumination conditions and glass occlusion. We take the advantage of dataset's small size and its diversiform conditions to evaluate the adaptation of our approach and to find the optimized parameters set.

ORL Database: ORL database has 40 distinct subjects with 10 images each identity. Images are taken under varying head orientation, face expressions, illumination conditions and with or without glasses (see Figure 5.2).

PIE Database: There are 11554 images of 68 subjects in PIE database. Images of each subject are taken under different illumination conditions and head orientations (see Figure 5.3). Because of the huge amount of data, we employed GPU for experiments which also explore the potential of computing efficiency of Macropixel Approach using parallel computing.

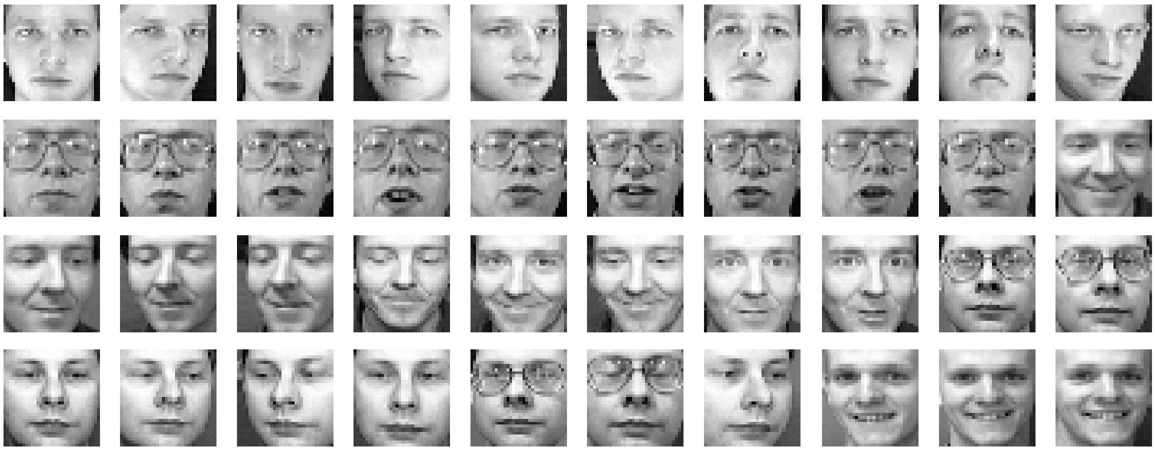


Figure 5.2: Face images of 4 sample subjects from ORL Database. Each subject contains 10 images with different illumination conditions, face directions and expressions.

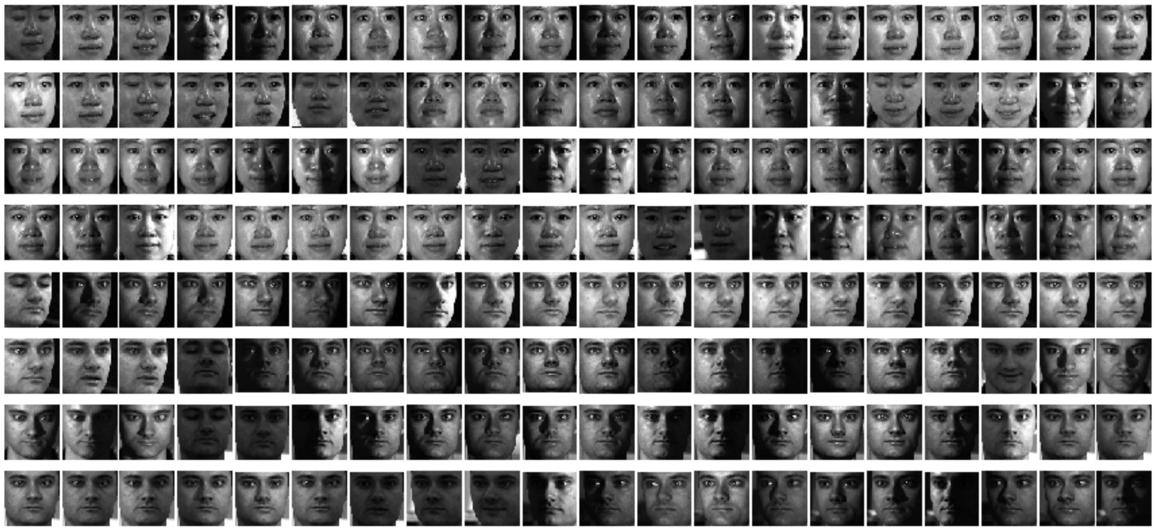


Figure 5.3: Face images of 2 sample subjects from PIE Database. Figure only shows 80 images for each subject. There are totally 68 subjects and each of them contains more than 100 images.

Table 5.1: The parameter settings of weighted filter experiment for Yale Database.

Type of Parameter	**Parameters Set**
Size of Image($N \times M$)	$N \times M = \{32 \times 32, 64 \times 64^{\star}\}$
Template Images per Subject(NT)	$NT = \{nt \vert 2 \leq nt \leq 8, nt \in Z\}$
Weight Training Images per Subject(WT)	$WT = \{wt \vert 1 \leq wt \leq NT - 1, wt \in Z\}$
Training Frequency(X)	$X = \{25, 50, 100^{\star}, 200\}$
Independent Training Phase	$\{NULL, I_{Yale_{wt}^{nt}}\}^{\diamond}$

Note:

$\star$ indicates default setting which is applyed for experiments testing other type of parameters.

$\diamond$: 'NULL' indicates experiment without independent training. $nt = 2, 3, 4, 5, 6, 7, 8, mt = 2, ..., nt - 1$

5.2 Experiment Design

The designed experiment is consisting of three phases which test the performance of our proposed system integrally or separately. As overlapping and weighted filter could operate independently, we apply each of them into the recognition for analyzing and combine them together for optimized results.

5.2.1 Experiment for Weighted Filter

To evaluate the performance of weighted filter, we start with testing the system on Yale database with a wild range of different parameters sets. The parameter settings are listed in Table 5.1.

The weight training experiment traverses through all the WT settings in order to

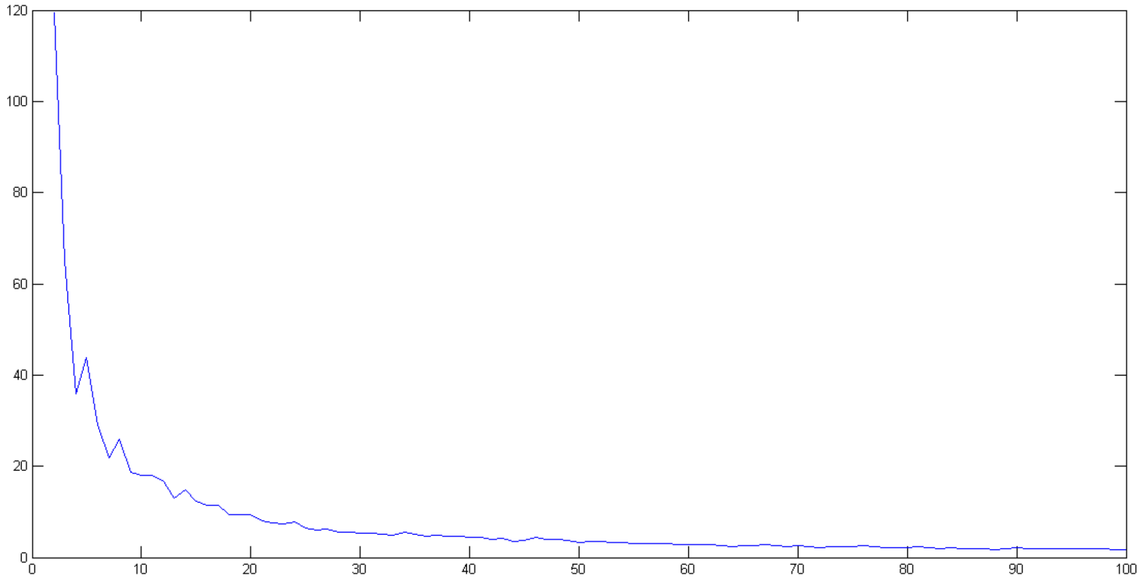


Figure 5.4: The total value change of weighted filter over training times. It drops under 5 after repeating 50 times.

pick out the WT number which could mostly improve the performance of the system and avoid over fitting or under fitting. We test on weight training frequency to balance efficiency and effectiveness as the frequency directly affects time consumption of the system. The change of weighted filter over training times is shown in Figure 5.4. The difference value has decreased to a slight degree after 50 times of repeating, which is the basis of our parameter selections for experiment on training frequency.

We also study performance of independent weight training in the experiment. After randomly choosing one data sequence in Yale, we train the filter with every NT-WT combination and use them on Yale face recognition to find the best training parameters. As face images in different databases are taken under various conditions, the filter trained by Yale is then employed to be tested on ORL and PIE to see if it could improve performance of the system.

Table 5.2: The parameter settings of overlapping experiment for Yale, ORL and PIE Databases.

Type of Parameter	**Parameters Set**
Size of Image($N \times M$)	$N \times M = \{32 \times 32, 64 \times 64\}$
Template Images per Subject(NT)	$NT = \{nt \vert 2 \leq nt \leq 8, nt \in Z\}$
Stride of Overlap(S)	$S = \{1, 2\}$

Note:

⋆ indicates default setting which is applyed for experiments testing other type of parameters.

5.2.2 Experiment for Overlapping

As overlapping works better on larger dataset, we have tested on each database with different overlap degrees. The settings are shown in Table 5.2. The optimized parameter setting will be applied to the compact system experiment.

Considering our experiment are based on macropixels of size 4 × 4, all three databases are tested for overlapping with a stride of 1 or 2.

5.2.3 Experiment for Compact System

Table 5.3 shows the parameter settings for compact system experiment. For compact system, we choose the optimized parameter setting for testing. The parameters we selected are obtained by using the trial and error approach.

Table 5.3: The Parameter Settings of compact system experiment for Yale, ORL and PIE Databases.

Type of Parameter	**Parameters Set**
Size of Image($N \times M$)	$N \times M = \{32 \times 32, 64 \times 64\}$
Template Images per Subject(NT)	$NT = \{nt \| 2 \leq nt \leq 8, nt \in Z\}$
Weight Training Images per Subject(WT)	$WT = 1$
Training Frequency(X)	$X = 100$
Stride of Overlap(S)	$S = \{1, 2\}$
Shifting Pixel Number	2
Independent Training Phase	$\left\{NULL, I_{Yale}{}_1^2\right\}^\diamond$

5.3 Experiment Result

5.3.1 Result for Weight Training

Table 5.4 and 5.5 exhibit the recognition error rate of Macropixel Comparison system implanted with Weighted Filter Counter for Yale Database. 5.4 lists performances of all the WT settings and 5.5 lists performances of system employing independent weight training. Results from both tables are tested on images with size 64 × 64. As shown in the results, the optimal parameter settings for weighted filter is $WT = 1$ and for independent training it is $I_{Yale}{}_2^1$. As number of training images or number of template image for independent training gets higher, the error rate ascends. Comparing to the original result, the error rate has decreased by nearly 40

Table 5.6 and 5.7 show the results for other parameter settings, which are 32 × 32 image size and training frequency(X) separately. Weighted Filter also increases the performance of recognition for 32 × 32 images as average error rates fall by nearly 30

Table 5.4: Results of Macropixel approach with Weighted Filter for Yale database.

Training Images	**Template Images(NT)**						
(WT)	2	3	4	5	6	7	8
original	14.9420	11.0306	8.0270	6.1481	4.6800	5.2722	3.4519
1	**10.3111**	**7.3333**	**4.9524**	**3.8444**	**2.8800**	**3.4000**	**1.7778**
2	NULL	9.0000	5.2000	4.0000	2.9333	3.5667	1.9556
3	NULL	NULL	5.2762	4.1556	3.0133	3.6333	2.0444
4	NULL	NULL	NULL	4.1778	3.0667	3.7333	2.0000
5	NULL	NULL	NULL	NULL	3.0667	3.8667	2.0444
6	NULL	NULL	NULL	NULL	NULL	3.8333	2.0889
7	NULL	NULL	NULL	NULL	NULL	NULL	2.0444

Table 5.8 and Table 5.9 contain the results from ORL and PIE with weight training. The improvement is not as remarkable as for Yale, but it could still benefits the recognition rates for both sizes.

5.3.2 Results for Overlap

Table 5.10 and 5.11 show the experiment results on all three database testing overlapping. The results of 5.10 shows that most the recognition rate does not get better but a little bit worse for Yale by employing overlapping because of a small quantity of subjects except the one from image size of 32×32 with a stride of 2. As seen from the results, the performance of system with overlap($S = 2$) is better than deep overlap ($S = 1$). And it is worth noting that along with rising number of template images (NT), the performance of system is getting better.

For larger databases which have more subjects, performance of the system has

Table 5.5: Results of Macropixel approach with Independent Weighted Filter for Yale database.

Independent Train Operator	**Template Images(NT)**						
$(\boldsymbol{I_{DB_{nt}^{wt}}})$	2	3	4	5	6	7	8
original	14.9420	11.0306	8.0270	6.1481	4.6800	5.2722	3.4519
$\boldsymbol{I_{Yale_2^1}}$	**9.3037**	**6.8000**	**4.7238**	**3.6889**	**2.8000**	**3.1667**	**1.9556**
$I_{Yale_3^1}$	10.9778	7.6500	5.3143	4.0667	3.2533	3.5333	2.0444
$I_{Yale_3^2}$	10.9926	7.7333	5.4095	4.2444	3.2000	3.6000	2.0889
$I_{Yale_4^1}$	10.9037	7.8000	5.4286	4.1778	3.2000	3.8333	2.0889
$I_{Yale_4^2}$	11.0815	7.8667	5.5238	4.2000	3.1467	3.9667	2.0889
$I_{Yale_4^3}$	11.2815	8.0000	5.5619	4.2444	3.2533	4.0000	2.2667
$I_{Yale_5^1}$	11.2444	7.8667	5.6381	4.3778	3.2800	3.7333	2.2222
$I_{Yale_5^2}$	11.4815	8.0667	5.7143	4.4444	3.3067	3.9333	2.2667
$I_{Yale_5^3}$	11.7037	8.3000	5.8095	4.5111	3.3600	4.0333	2.2667
$I_{Yale_5^4}$	11.7778	8.4167	5.8476	4.5556	3.4133	4.1000	2.3111
$I_{Yale_6^1}$	10.9481	7.8167	5.3524	4.2000	3.1467	3.8000	2.0444
$I_{Yale_6^2}$	11.0815	7.9500	5.4286	4.3778	3.1467	3.7667	2.0889
$I_{Yale_6^3}$	11.1111	7.9833	5.5429	4.3778	3.2800	3.8000	2.0889
$I_{Yale_6^4}$	11.2148	8.0333	5.4667	4.3333	3.2800	3.8000	2.0444
$I_{Yale_6^5}$	11.3630	8.3000	5.6381	4.3778	3.3067	4.0667	2.1333
$I_{Yale_7^1}$	11.5852	8.2667	5.7524	4.5111	3.4933	4.2333	2.2667
$I_{Yale_7^2}$	11.8667	8.5167	5.9238	4.6222	3.6267	4.3000	2.5333
$I_{Yale_7^3}$	11.9556	8.5000	5.9810	4.6889	3.6267	4.3000	2.4000
$I_{Yale_7^4}$	12.0148	8.5833	6.0952	4.7333	3.6533	4.3333	2.5333
$I_{Yale_7^5}$	12.1185	8.6500	6.0571	4.7778	3.7067	4.3333	2.5778
$I_{Yale_7^6}$	12.1630	8.7333	6.0762	4.8000	3.7867	4.3667	2.5778
$I_{Yale_8^1}$	11.2593	7.9667	5.5238	4.4444	3.2267	3.8333	2.2222
$I_{Yale_8^2}$	11.4667	8.1500	5.7143	4.5111	3.4400	4.0333	2.2667
$I_{Yale_8^3}$	11.6296	8.2167	5.7905	4.5333	3.5200	4.1000	2.3111
$I_{Yale_8^4}$	11.7926	8.3333	5.9238	4.5333	3.5200	4.2000	2.3556
$I_{Yale_8^5}$	11.8963	8.5333	5.9619	4.5778	3.5467	4.2333	2.3111
$I_{Yale_8^6}$	11.8963	8.5000	5.9238	4.5778	3.5733	4.2000	2.3111
$I_{Yale_8^7}$	12.1037	8.6833	6.0381	4.5778	3.5467	4.2333	2.3111

Table 5.6: Results of smaller size images experiment for Yale database.

Training Condition	**Template Images(NT)**						
	2	3	4	5	6	7	8
original	24.9267	19.3750	16.2781	14.0241	13.7436	12.6750	11.6529
$WT = 1$	**17.1630**	**13.2833**	**10.3238**	**8.8222**	**8.0933**	**7.3667**	**6.5333**
$I_{Yale}{}^{1}_{2}$	21.4074	16.3417	13.8095	11.5778	11.1733	10.7667	9.6000

Table 5.7: Results of frequency experiment for Yale database.

Training Frequency(X)	**Template Images(NT)**							**Total Difference**
	2	3	4	5	6	7	8	
original(100)	10.3111	**7.3333**	**4.9524**	3.8444	2.8800	**3.4000**	**1.7778**	0
25	10.3333	7.4333	5.0667	**3.8222**	**2.8000**	3.4667	1.8222	-0.2454
50	10.2889	7.4167	5.0095	3.8444	2.8267	3.4000	1.8667	-0.1539
200	**10.2667**	7.3833	4.9905	3.8889	2.8267	3.4333	1.8667	-0.1571

Table 5.8: Results of Macropixel approach with Independent Weighted Filter for ORL database.

Image Size	**Training Condition**	**Template Images(NT)**						
		2	3	4	5	6	7	8
32 × 32	original	13.2620	7.2505	3.6090	1.9380	1.4038	0.7722	0.7583
	$I_{ORL}{}^{1}_{2}$	**13.0875**	**6.8786**	**3.4417**	**1.8200**	**1.2125**	**0.5667**	**0.7250**
64 × 64	original	9.6188	5.1643	2.4083	1.1600	0.8250	0.4167	0.3500
	$I_{ORL}{}^{1}_{2}$	**9.1875**	**5.0250**	**2.2333**	**1.0400**	**0.6875**	**0.3500**	**0.3250**

Table 5.9: Results of Macropixel approach with Independent Weighted Filter for PIE database.

Image Size	**Training Condition**	**Template Images(NT)**									
		5	10	20	30	50	70	80	90	110	130
32 × 32	original	32.8474	17.3366	7.0250	3.9663	1.6895	0.8564	0.6998	0.5469	0.3378	0.2106
	$I_{PIE_2^1}$	**23.0199**	**10.3164**	**3.7177**	**1.9252**	**0.7808**	**0.3830**	**0.3255**	**0.2429**	**0.1620**	**0.1002**
64 × 64	original	21.3626	9.7984	3.7438	2.1174	1.1228	**0.6589**	0.5920	0.46040	0.3933	0.2917
	$I_{PIE_2^1}$	**18.9099**	**8.6086**	**3.3710**	**1.9529**	**0.9973**	0.6597	**0.5411**	**0.4233**	**0.3343**	**0.2572**

significantly improved. For ORL, implanting deep overlap which is overlap with stride of 1 has decreased the error rate by more than 30 percent. As for PIE database, the situation is similar as the average error rate decreases by 40 percent on average.

5.3.3 Results for Compact System

By designing and conducting a series of separated experiments and exploring the characteristics of our proposed system, we are able to select the optimal parameters for face recognition. Table 5.12 and 5.13 show the compact system results for all three databases. Results from Yale for compact system are not as good as the ones implementing weighted filter individually, as overlapping works better on database with more subjects.

Performance of the compact system on ORL and PIE have exceeded the ones of original and separated system. We also test weighted filter trained by Yale database on ORL and PIE. Although the recognition rates are not as good as the optimal results, it is still better than the original ones. These results implies that weighted filter is of reasonable applicability and by producing pre-trained weighted filter and

Table 5.10: Results of Overlapping for Yale and ORL database.

Target Database	**Image Size**	**Stride (S)**	**Template Images(NT)**						
			2	3	4	5	6	7	8
Yale	32×32	original	24.9267	19.3750	16.2781	**14.0241**	13.7436	12.6750	11.6529
		$S = 2$	**24.4321**	**18.9778**	**15.6540**	14.0778	**13.6311**	**12.5778**	**10.8519**
		$S = 1$	25.5481	19.6000	16.0571	14.6556	13.9200	12.8833	11.9556
Yale	64×64	original	**14.9420**	**11.0306**	**8.0270**	6.1481	4.6800	5.2722	3.4519
		$S = 2$	15.5370	11.8542	8.9286	**5.4815**	**4.0667**	**5.0417**	**3.2222**
		$S = 1$	16.7704	12.3500	9.4381	7.1630	5.0000	5.6333	3.3333
ORL	32×32	original	13.2620	7.2505	3.6090	1.9380	1.4038	0.7722	0.7583
		$S = 2$	9.6594	4.6399	1.8597	**0.8250**	0.5271	0.2833	0.2375
		$S = 1$	**9.6521**	**4.5929**	**1.7306**	0.8850	**0.4875**	**0.1917**	**0.1375**
ORL	64×64	original	9.6188	5.1643	2.4083	1.1600	0.8250	0.4167	0.3500
		$S = 2$	8.8031	4.3631	1.6750	0.8000	0.5188	0.3667	0.3750
		$S = 1$	**8.0062**	**3.8786**	**1.5250**	**0.6900**	**0.4000**	**0.3000**	**0.3500**

Table 5.11: Results of Overlapping for PIE database.

Image Size	**Stride (S)**	**Template Images(NT)**									
		5	10	20	30	50	70	80	90	110	130
32×32	original	32.8474	17.3366	7.0250	3.9663	1.6895	0.8564	0.6998	0.5469	0.3378	0.2106
	$S = 2$	**23.2225**	**9.4408**	**2.9597**	**1.3428**	0.4298	0.2283	0.1646	0.1301	0.0672	0.0445
	$S = 1$	23.8812	9.7518	3.0602	1.3805	**0.4055**	**0.1698**	**0.1076**	**0.0799**	**0.0447**	**0.03135**
64×64	original	21.3626	9.7984	3.7438	2.1174	1.1228	0.6589	0.5920	0.46040	0.3933	0.2917
	$S = 2$	17.1665	6.8520	2.2677	1.1868	0.5762	0.3881	0.3191	0.2554	0.1987	0.1608
	$S = 1$	**16.2967**	**6.2375**	**1.9732**	**0.9869**	**0.4876**	**0.3095**	**0.2306**	**0.2156**	**0.1558**	**0.1197**

Table 5.12: Optimized results of Compact System for Yale and ORL.

Target Database	**Image Size**	**Stride (S)**	**Training Condition**	**Template Images(NT)**						
				2	3	4	5	6	7	8
Yale	32×32	$S = 0$	Original	24.9267	19.3750	16.2781	14.0241	13.7436	12.6750	11.6529
		$S = 2$	$I_{Yale_2^1}$	**20.9778**	**15.7500**	**12.9143**	**11.8222**	**10.7467**	**10.3000**	**8.6667**
Yale	64×64	$S = 0$	Original	14.9420	11.0306	8.0270	6.1481	4.6800	5.2722	3.4519
		$S = 2$	$I_{Yale_2^1}$	**10.0593**	**7.1500**	**4.9333**	**3.7778**	**2.8000**	**3.2667**	**2.1778**
ORL	32×32	$S = 0$	Original	13.2620	7.2505	3.6090	1.9380	1.4038	0.7722	0.7583
		$S = 1$	$I_{ORL_2^1}$	**9.2813**	**4.3214**	**1.6333**	**0.7400**	**0.3875**	**0.1833**	**0.1000**
ORL	64×64	$S = 0$	Original	9.6188	5.1643	2.4083	1.1600	0.8250	0.4167	0.3500
		$S = 1$	$I_{ORL_2^1}$	**7.9688**	**3.7571**	**1.4833**	0.6600	**0.3875**	0.2833	0.2500
		$S = 1$	$I_{Yale_2^1}$	8.1687	3.8429	1.4917	**0.6200**	0.4000	**0.1833**	**0.1500**

Table 5.13: Optimized results of Compact System for PIE.

Image Size	**Stride (S)**	**Training Condition**	**Template Images(NT)**									
			5	10	20	30	50	70	80	90	110	130
32×32	$S = 0$	Original	32.8474	17.3366	7.0250	3.9663	1.6895	0.8564	0.6998	0.5469	0.3378	0.2106
	$S = 1$	$I_{PIE_2^1}$	**21.1784**	**8.4328**	**2.6290**	**1.1993**	**0.3586**	**0.1540**	**0.1050**	**0.0762**	**0.0442**	**0.0287**
64×64	$S = 0$	Original	21.3626	9.7984	3.7438	2.1174	1.1228	0.6589	0.5920	0.46040	0.3933	0.2917
	$S = 1$	$I_{PIE_2^1}$	**14.0806**	**4.4878**	**1.4028**	**0.7883**	**0.3311**	**0.2366**	**0.1963**	0.1944	**0.0982**	**0.0974**
	$S = 1$	$I_{Yale_2^1}$	15.3077	5.7477	1.7628	0.86959	0.4325	0.2772	0.2169	**0.1907**	0.14482	0.0983

reuse it on other occasion, we can reduce the computation and time significantly.

5.4 Parallel Computing Efficiency

In order to exhibit the efficiency improvement of our system utilizing general purpose computation based on GPU, we test operation time on PIE database which is shown in Figure 5.5. The experiment records the operation time of one comparison operation comparing 221000 macropixels of size 4×4 with 2714 ones. The experiment results are obtained on a Nvidia Geforce GTX Titan Black 6GB GPU and Intel Xeon

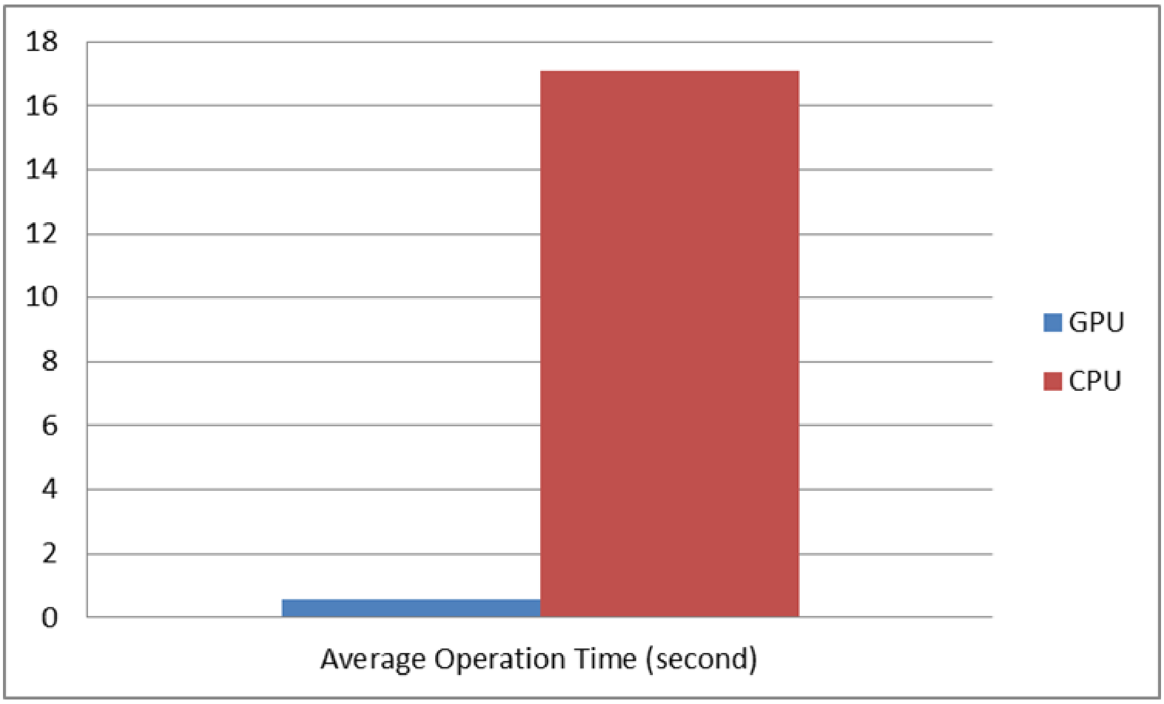


Figure 5.5: Red bar is the running time using CPU with an average of 17 seconds per comparison operation. Blue bar is the running time using GPU with an average of 0.58 seconds per operation.

E5-2603 1.8 Ghz CPU. As PIE has more than 10,000 images, the operation(basically matrix multiplication) requires a significant amount of computing resources. We can see from the figure that Parallel computing employing GPU accelerates the computation up to 30 times.

Chapter 6

Conclusion and Discussion

6.1 Conclusion

In this thesis, we introduce a face recognition approach named Convolutional Macropixel Approach which is developed based on the naive Macropixel Comparison Method and inspired by the Convolutional Neural Network Architecture.

Two major improvements are introduced to the approach and tested on several Datasets(i.e., Yale, ORL and PIE) separately or together. From what we learn from the experiments, the Convolutional Macropixel approach achieves significantly better recognition rate than the original Macropixel Comparison method in all three datasets. Weighted Filter is a pre-trained filter which is connected to the recognition phase and could generally increase the recognition rate. It works even better with less template images and less subjects. Overlap especially deep overlap uses convolutional way to compare macropixels and could improve the recognition rate significantly when the number of recognition subjects is enormous.

Parallel Computation is also tested for the proposed method and could greatly improve the recognition efficiency when dealing with large amount of data.

Research proves that the idea of introducing CNN framework into mathematical face recognition method is feasible. It also demonstrates the potential of Macropixel Comparison Approach which not only could benefit from parallel computing, but also could utilize pre-trained weighted filter.

6.2 Future Work

The conception of further improvements for the approach is listed below:

- Face identification. The approach is proved to be reliable on face verification. To better explore the applicability of Convolutional Macropixel Approach, we could introduce it to face identification.
- Weight training. It is possible to utilize other weighted filter training methods including deep learning framework to the approach.
- Preprocessing. Inherited from the original macropixel approach, any preprocessing methods that applied to the images are compatible with our imrpoved approach.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Computer vision-eccv 2004*, pages 469–481, 2004.

[2] M. Ballantyne, R. S. Boyer, and L. Hines. Woody bledsoe: His life and legacy. *AI Magazine*, 17(1):7, 1996.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[4] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[5] A. Bukis and R. Simutis. Face orientation normalization using eye positions. *Computer Technology and Application*, 4(10), 2013.

[6] L. Chen. Pairwise macropixel comparison can work at least as well as advanced holistic algorithms for face recognition. In *BMVC*, pages 1–11. Citeseer, 2010.

[7] D. Cox and N. Pinto. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In *Automatic Face & Gesture Recogni-*

tion and Workshops (FG 2011), 2011 IEEE International Conference on, pages 8–15. IEEE, 2011.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[9] J. G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10):847–856, 1980.

[10] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[11] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1408–1423, 2004.

[12] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

[13] B. Heisele, M. Pontil, et al. Face detection in still gray images. Technical report, DTIC Document, 2000.

[14] R. Heitmeyer. Biometric identification promises fast and secure processing of airline passengers. *ICAO journal*, 55(9):10–11, 2000.

[15] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[16] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[17] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

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

[19] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on computers*, 42(3):300–311, 1993.

[20] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[21] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[24] C. Liu. A bayesian discriminating features method for face detection. *IEEE transactions on pattern analysis and machine intelligence*, 25(6):725–740, 2003.

[25] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman. Sex, mixability, and modularity. *Proceedings of the National Academy of Sciences*, 107(4):1452–1457, 2010.

[26] R. M. Makwana. Illumination invariant face recognition: a survey of passive methods. *Procedia Computer Science*, 2:101–110, 2010.

[27] F. Matt Hicks. Making photo tagging easier, 2011. Online; accessed 01-February-2018.

[28] C. N. Matthew Braga. Facial recognition technology is coming to canadian airports this spring, 2017. Online; accessed 06-March-2018.

[29] Microsoft. Windows hello face authentication, 2016. Online; accessed 10-August-2016.

[30] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[31] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pages 130–136. IEEE, 1997.

[32] A. J. O'Toole, D. A. Roark, and H. Abdi. Recognizing moving faces: A psychological and neural synthesis. *Trends in cognitive sciences*, 6(6):261–266, 2002.

[33] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[34] P. J. Phillips, W. T. Scruggs, A. J. OToole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. Frvt 2006 and ice 2006 large-scale results. *National Institute of Standards and Technology, NISTIR*, 7408(1), 2007.

[35] K. Ramírez-Gutiérrez, D. Cruz-Pérez, and H. Pérez-Meana. Face recognition and verification using histogram equalization. In *Proceedings of the 10th WSEAS international conference on Applied computer science*, pages 85–89. World Scientific and Engineering Academy and Society (WSEAS), 2010.

[36] M. Rätsch, S. Romdhani, and T. Vetter. Efficient face detection by a cascaded support vector machine using haar-like features. In *Joint Pattern Recognition Symposium*, pages 62–70. Springer, 2004.

[37] S. Romdhani, P. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 695–700. IEEE, 2001.

[38] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

[39] J. Ruiz-del Solar and J. Quinteros. Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different preprocessing approaches. *Pattern Recognition Letters*, 29(14):1966–1979, 2008.

[40] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[41] J. C. Russ. *The image processing handbook*. CRC press, 2016.

[42] M. S. Sarfraz, O. Hellwich, and Z. Riaz. Feature extraction and representation for face recognition. In *Face Recognition*. InTech, 2010.

[43] 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*, pages 815–823, 2015.

[44] S. Shan, W. Gao, B. Cao, and D. Zhao. Illumination normalization for robust face recognition against varying lighting conditions. In *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, pages 157–164. IEEE, 2003.

[45] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.

[46] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

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

[48] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[49] 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*, pages 1701–1708, 2014.

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

[51] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001.*

Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–I. IEEE, 2001.

[52] C. Von Der Malsburg. The correlation theory of brain function. In *Models of neural networks*, pages 95–119. Springer, 1994.

[53] P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[54] L. Wiskott, N. Krüger, N. Kuiger, and C. Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779, 1997.

[55] 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*, 31(2):210–227, 2009.

[56] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1):34–58, 2002.

[57] C. Zhang and Z. Zhang. A survey of recent advances in face detection, 2010.

[58] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma. Face recognition with contiguous occlusion using markov random fields. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1050–1057. IEEE, 2009.