A novel method of face verification based on EM algorithm

A Thesis presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Ran Pan

Dr. Xu Han, Thesis Supervisor

December 2014

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

A NOVEL METHOD OF FACE VERIFICATION
BASED ON EM ALGORITHM

presented by Ran Pan,
a candidate for the degree of Master and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Xu Han

Dr. Ye Duan

Dr. Zhihai He

ACKNOWLEDGMENTS

This research project would not have been possible without the support of many people. First, I wish to express my gratitude to my supervisor Prof.Han who was abundantly helpful and offered invaluable assistance. His insight lead to the original proposal to do the face verification in LFW and help me finding a novel way to express face model. Then, deepest gratitude are also result from the members of the supervisory committee who are both very experienced with face recognition, Prof.Duan and Prof.He, who would like to give me some great suggestions.

Special thanks also to all my graduate friends: Zhixin Ren, HaiPei Fan, Guang Chen and Miao Sun for sharing the novel ideas, literature and invaluable assistance. Specially I want to thank Arthur for his selfless with academically and living help.

I would also like to extend my appreciation to Tom Gittermeier and Gates, I got much knowledge of programming and software by their guidance. Additionlly, Moyi Zhang, Zhaolong Zhong and Li Li give me much advice about computer vision and statistic methods.

Finally I would like to extend my deepest gratitude to my parents Xiaochuan Pan and Qi chen without whose love, support and understanding I could never have completed this master degree.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

In this paper, we implement a novel joint Bayesian method based on the classical Bayesian face recognition method by Baback Moghaddam et al and a creative paper ”Bayesian Face Revisited: A Joint Formulation”. One face is divided into two parts by us: identity and variation, which results a much better performance than the prior algorithms and the verification rate reaches 93 % on LFW.

To compare each parameters in EM algorithm, we use two types training ways and add a validation set as the stopping criterion. Additionally, we also reduce the computational complexity by changing log likelihood ratio into a closed form. These changes make our algorithm outweigh the performance of the original joint Bayesian method with even lower dimensions LBP feature.

Key words: LBP, EM, Joint Bayesian

Chapter 1

Introduction

1.1 Motivation

Face recognition is a ubiquitous and difficult topic. It can be divided into two branches: face verification and face identification. On the one hand, face verification is to ask Are they the same person? On the other hand, face identification is about who is he? The former one is used more wide because it does not need much amount of data which the other one does. If we want to know who he is from the face image, we must label his face for train. However, to decide two faces if they are the same subject requires much less information. Therefore, the challenge of face verification is to test 100 people by training only 10 of them.

In recent years, big data has been going through all the fields. More and more projects have to solve the problem that how to deal with the huge data warehouse to keep all procedures running fast and stable. Face verification also faces some similar problems[1][2]: we can get a high dimension feature for a face image but it may take too much time to train and test. Generally, to reduce the dimension of features[3] or improve the hardware of projects are two solutions[4].

Many state of arts algorithms focus on looking for new features and mixing some prior algorithms. It cannot be denied that they are both good ideas and more easier to improve performance directly, though most of them are difficult to reused. From a totally different viewpoint, we find a joint Bayesian model to outperform the other methods, which can also be used flexible.

1.2 Previous Work

Before the 1990s, most research about face recognition is based on features and the face recognition systems became possible such as the layered neural network system of O'Toole et al[5]. In the 1990s, as a humans unique feature, facial feature drew more scholars attention. Low recognition rate made the people feel frustrated and someone propose to use template matching, which led to the famous eigenface technique using Principle Component Analysis (PCA) by M. Turk and A. Pentland [6].

Many newest methods were originated from the eigenface at that time. They extended the eigenspaces to subspace learning way and showed a more powerful result than standard eigenface, such as Etemad and Chellappas, which used Linear Discriminant Analysis (LDA)[7]. Besides, with the recognition rate increasing, the applications of face recognition system also attracted the companies about security or army. For instance, Craw modeled the shape of the face named ”shape-free” face and combined it with the other methods[8]

As mentioned above, many successful features have been generally applied in face recognition. Sift is invariant to image scale and rotation, and provide robust matching across a substantial range of afne distortion, change in 3D viewpoint, addition of noise, and change in illumination[9]. Histogram of Oriented Gradients (HOG) is similar to that of scale-invariant feature transform descriptors but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy[10]. Gabor filter is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination[11]. LBP is the particular case of the Texture Spectrum model proposed in 1990 and it is widely used in face recognition[12].

Learning-based descriptor (LE) is a novel representation to solve the face matching issue[13]. Recent year, the face data have been mined more deep and the face data system also become more robust, which has been divided to a pipeline with 3 different fields in an order: face detection, face alignment and face verification (recognition)[14].

Face detection is obviously the first step in the whole system because we need to get the exact face regions[15]. However, the face region also contains two much information, especially a person with different angles and emotions. To avoid the big data from face, the researchers begin using face alignment method to extract some points on the face to represent the whole face. This means if we want to get LBP from a face, only the alignment points need to be computed[16]. Nowadays, there are still many algorithms pick the features from the whole face directly, which will result in a long training time, more memories of hardware cost and redundant information. But the performance of them may not lower than the algorithms using face alignment so that this procedure is often ignored by some researchers. The last step is face verification or recognition algorithms that can tell us the identity[17].

Besides, choosing a better face image dataset to train and test is essential. Yale Face Database is very famous but it is too small. It contains only 165 grayscale images in GIF format of 15 individuals. Though the extended Yale Face Database B is a large dataset with 16128 images, there are only 28 human subjects under 9 poses and 64 illumination conditions[18]. Actually, they are pretty good datasets in the beginning because the training set is diversity. Later, the researchers found that it is not too difficult to get a good result in these databases. More importantly, the face recognition algorithm performs still not well when it is used in a real application even its recognition rate has reached 99% on Yale Face Database. Therefore, people have realized two things: the face images should be got not only from the cameras in the lab but also outside to make the algorithms more robust; more human subjects will result in much more problems and we need a wide and deep database. Recent years, Labeled Faces in the Wild (LFW)[19], built by UMASS Computer Vision Laboratory, is used widely and become the most publicly known dataset. There are a total 5749 people with13233 images and 1680 of them with 2 or more than 2 images.

1.3 Our Approach

Our approach utilizes much newest technology and creates a new model based on Normal distribution and Bayesian face recognition. So the face verification becomes a binary problem. The classical Bayesian methods use the difference between two images as the features to decide if they are the same person. Instead, we model two faces jointly with an appropriate prior as the face representation.

Obviously, the similarity between two images should be tested from two images but it will lose much useful information if we only compute difference between two images as features in the training step. There are some ways that can just relieve this disadvantage but not solve it basically, such as metric learning. However, our model is directly from the features of images but not the similarity of the pairs. It solves the above problem that results from using the data after processing.

To get a convincing result, we need to use a good and popular dataset. As mentioned above, LFW owns more than 5000 subjects of face images, it is very appropriate for the face verification. By the doing many experiments, the verification rate of our algorithm has reached about 90% on LFW.

Chapter 2

Face Model

2.1 Motivation

Let H_I represents the intra-pairs that two faces f_1 and f_2 belong to the same subject, and H_E represents the extra-pairs that two faces are from different subjects. Then, the face verification problem focus on the similarity of intra-pairs and extra-pairs based on the MAP (Maximum a Posterior) rule. To get the decision, we test a log likelihood ratio r(f_1, f_2).

$$r(f_1, f_2) = \log \frac{P(f_1, f_2|H_I)}{P(f_1, f_2|H_E)}. \tag{2.1}$$

Equation.2.1 is a common measurement between two faces f_1 and f_2. In Bayesian face recognition[20] by Moghaddam et al, two conditional probabilities are Gaussian models used for model learning. The Bayesian face attracts more attention for its excellent performance. For example, Gabor filter is used to replace normal features to show the face difference[21]. Wang and Tang[22] partitions the face difference into three subspaces: intrinsic difference, transformed difference and noise. Instead of using a native Bayesian classifier, Li, Z. and Tang use a novel way to train by SVM[23].

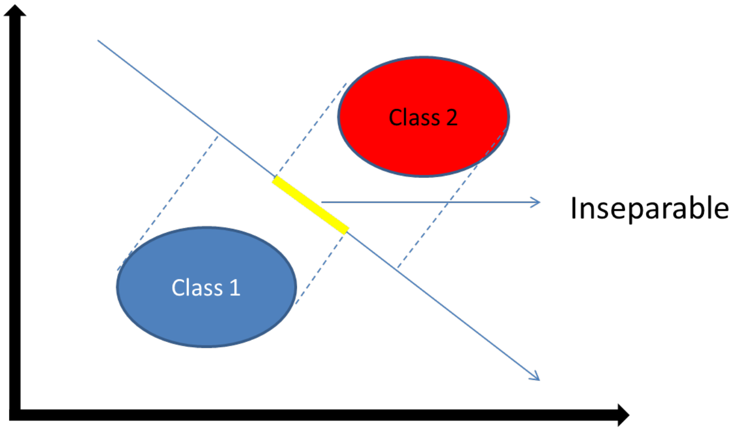


Figure 2.1: The 2-D data is projected to 1-D by xy.

As shown in Fig.2.1[17], the two classes in joint representation are inseparable after projecting the 2-D data to 1-D data, which means much information will be lost. Class1 and Class2 could be considered as an intra-personal and an extra-personal hypothesis in face recognition. Our method uses a competitive joint distribution and Bayesian framework based on a Gaussian model. We introduce an appropriate prior on face representation: each face is the summation of two independent Gaussian latent variables, i.e., intrinsic variable for identity, and intra-personal variable for within-person variation.

2.2 Joint Formulation

In this section, we will introduce the original joint formulation before our joint formulation. There are two mainly advantages for our new face model. Firstly, we directly use the original face images but not the difference of each pair, which avoid losing their common features. Secondly, we assume each person is composed of identity and variation. And all the identities belong to a Gaussian distribution; all the variations belong to the other one.

2.2.1 Original Models

As we talked in the introduction section, the face verification problem goes up later than the face recognition. People have recognized that the face verification can be used in much security application because it can work very well by training only a small dataset. Therefore, some researchers used the face model of recognition at the beginning.

In the LFW, some original face model is definitely classified by different subjects[24], it is a normal way in face recognition approach but it is not appropriate for face verification because it only emphasize the difference between all the training human subjects. It results in a narrow application: the system use this face model can only verify the people in training set. Then, people realized that the face verification is a very different field, the popular face model used in face verification appears: from the two Gaussian functions

$$\begin{aligned} P(f_1, f_2|H_I) &= N(0, \sigma_I), \\ P(f_1, f_2|H_E) &= N(0, \sigma_E) \end{aligned} \tag{2.2}$$

we often utilize the similarity of each pair, where σ_I and σ_E can be estimated from the intra-personal pairs and extra-personal pairs respectively. At the test time, the log likelihood ratio between two probabilities is used as the similarity metric.The main disadvantage of this face model is the "Separately". First, each human subject will be represented separately because the variations are regarded as independent. Second, it is difficult to compute each variation and it will lose some important information between different subjects.

As a whole, the original model focus on changing the features or the model learning way but ignoring the constitute of the model.[25][26][27][28]

Our new joint model ignore the restriction of each subject, we use identity part to emphasize the extra feature and the variation to emphasize the intra feature, which seems opposite to the original face models. A face can be represented by a sum of two independent Gaussian variables:

$$f = \mu + \epsilon \tag{2.3}$$

Where f is the observed face with the mean of all faces subtracted, I represents its identity is the face variation. These two variables belong to two Gaussian distributions with zero mean such as N (0, C_μ), N (0, C_ϵ) and C_μ ,C_ϵ are what we want to know. Because the means of these two Gaussian functions are 0, we can use only C_μ, C_ϵ to represent a face x. It is very useful to store the model and run the face recognition system in a real-time.

Eqn.2.3 can be transfer to a linear form and the independent assumption between I and , the covariance of two faces is:

$$cov(f_i, f_j) = cov(\mu_i, \mu_j) + cov(\epsilon_i, \epsilon_j), i, j \in \{1, 2\} \tag{2.4}$$

Therefore, for the verification problem, the two conditions can be showed as follow:

Under H_I hypothesis, if the two faces x_1 and x_2 are the same person, their identity I_1, I_2 will be the same and their intra-person variations ϵ_1, ϵ_2 will be independent[29]. From the knowledge of Gaussian model and joint distribution, we can get the covariance of the conditional probability P (f_1, $f_2|H_I$):

$$\begin{aligned} cov(\mu_i, \mu_j) &= \begin{pmatrix} C_\mu & C_\mu \\ C_\mu & C_\mu \end{pmatrix}, \\ cov(\epsilon_i, \epsilon_j) &= \begin{pmatrix} C_\epsilon & 0 \\ 0 & C_\epsilon \end{pmatrix}, \\ \sigma_I = cov(f_i, f_j) &= \begin{pmatrix} C_\epsilon + C_\mu & C_\mu \\ C_\mu & C_\epsilon + C_\mu \end{pmatrix}, \end{aligned} \tag{2.5}$$

Under H_E hypothesis, if the two faces f_1 and f_2 are the different person, their identity I_1, I_2 and their intra-person variations ϵ_1, ϵ_2 will be both independent. We can also get the covariance of the conditional probability P (f_1, $f_2|H_E$):

$$
\begin{aligned}
cov(\mu_i, \mu_j) &= \begin{pmatrix} C_\mu & 0 \\ 0 & C_\mu \end{pmatrix}, \\
cov(\epsilon_i, \epsilon_j) &= \begin{pmatrix} C_\epsilon & 0 \\ 0 & C_\epsilon \end{pmatrix}, \\
\sigma_E = cov(f_i, f_j) &= \begin{pmatrix} C_\epsilon + C_\mu & 0 \\ 0 & C_\epsilon + C_\mu \end{pmatrix},
\end{aligned}
\tag{2.6}
$$

The 2-dimension matrix of covariance not only divides the two conditions exactly, but also integrates the identity part and variation part to a square matrix. More important is that we can use only one covariance of the model to control the result. We will talk about how to use this face model in next section.

Chapter 3

Algorithm

In our algorithm, Jian Sun's "Face Alignment via Component-based Discriminative Search"[16] is used to find the face alignment points. We get 100 points near eyes, nose and mouth for each face image, and implement LBP to get a 5900 dimension vector to represent a face. Our new face model will be trained by EM algorithm and the procedures are quite different with the one of normal Gaussian. Because our objective function is not used to compute our final result.

3.1 Feature

3.1.1 LBP

Local binary pattern (LBP)[30][31][32] is a very popular feature in face detection, alignment and recognition field. It is sensitive to texture and face wrinkle.

The Fig.3.1 shows a face image in LFW and its LBP feature, it is labeled as "Aaron

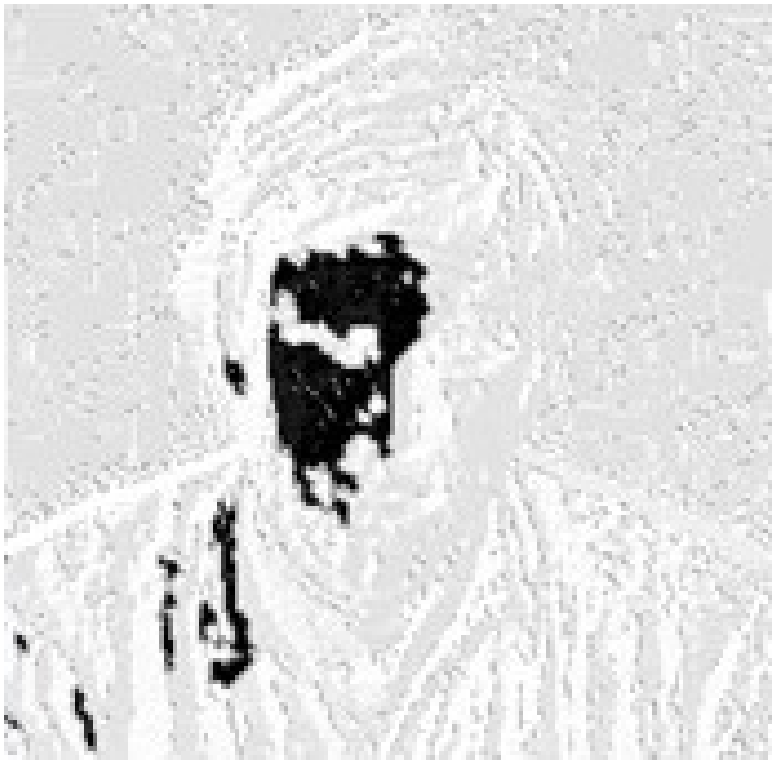


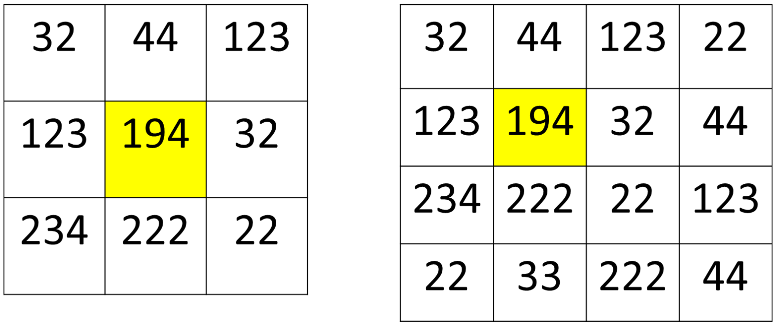
Figure 3.1: LBP of a face image

Eckhart". LBP describes the neighborhood of each point on the face. So there will be many types to implement LBP. We can use a circle or a square to restrict the neighbor region and set a different neighbor area size. We choose only eight points as the neighbor of a center pixel in our algorithm.

All the neighbor pixels will be compared to the center points. If a neighbor pixel is bigger than the center point, it will be 1. Otherwise, it will be 0. So, we can get an 8 bits binary number and change it to decimal number.

There are several ways to represent the LBP. For example, we can change the order of binary number and 00000110 will be 00110000 (48) in Fig.3.3.

Because we choose only 8 neighbor points, there will be $2^8 = 256$ patterns for a center point. However, the researchers found that only several types take more than



32	44	123
123	194	32
234	222	22

32	44	123	22
123	194	32	44
234	222	22	123
22	33	222	44

Figure 3.2: different neighbor size

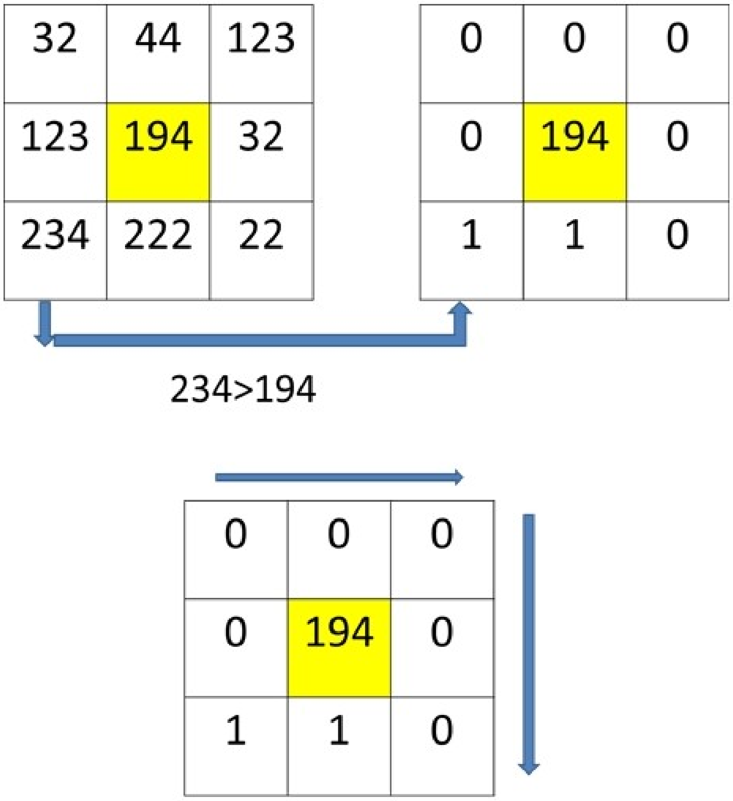
90 percent patterns. Ojala et al. define a new way to implement LBP using only one or two changing pattern: in a pattern, the number of changes of 0 to 1 or 1 to 0 is lower than twice[33].

Fig.3.4 is an example of uniform pattern and there are 58 uniform patterns in this restriction. The other 198 patterns are integrated into one pattern, which means final result contains 59 uniform patterns. For a face alignment point, we need to compute the LBP of its 21*21 neighbors to build a histogram. Based on every histogram of face alignment point, we can get a 5900-dimention feature for a 100 points face.

3.1.2 PCA

Principal Component Analysis (PCA)[34] is a statistics method using KarhunenLove transform (KLT) to change the correlated variables into the linearly uncorrelated variables. In our algorithm, 5900-dimention feature is too big to run fast due to the large memory and training data.

As Fig.3.5 shown to us, it will cost about 595MB internal storage if we use 5900-dimension feature directly. On the contrary, if we reduce the dimension to 800, it will take only 81MB. The experiments prove that the lower dimension feature some time



32	44	123
123	194	32
234	222	22

0	0	0
0	194	0
1	1	0

0	0	0
0	194	0
1	1	0

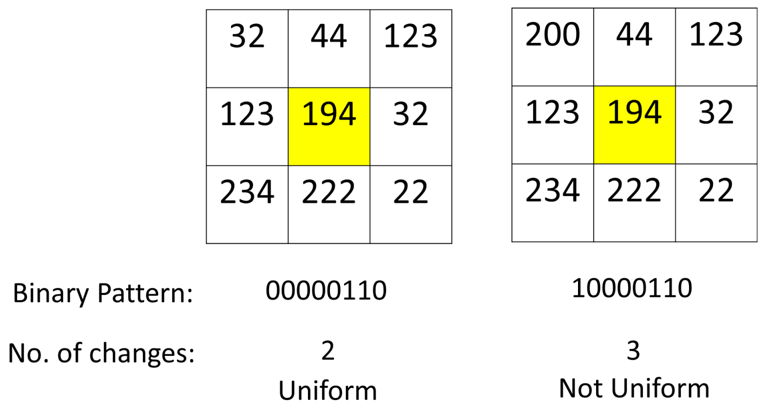
Figure 3.3: different representation

performs even better.

The main idea of PCA is to find the most important information and rank them in the order so that we can eliminate the less important ones and get the essence data from the samples.

Suppose the resolution of 13200 training images is 250*250 and the dimension of feature is 5900. First, we need to convert each image to a vector. Then, integrate each face vector into a face matrix with 13200*5900. And the procedures as follow:

Step 1 mean of the images:

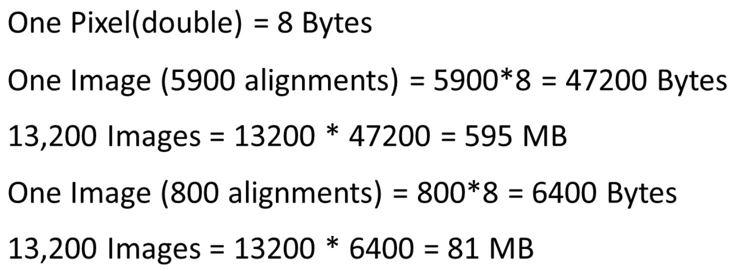


32	44	123
123	194	32
234	222	22

200	44	123
123	194	32
234	222	22

Binary Pattern:	00000110	10000110
No. of changes:	2 Uniform	3 Not Uniform

Figure 3.4: Uniform pattern of LBP



One Pixel(double) = 8 Bytes

One Image (5900 alignments) = 5900*8 = 47200 Bytes

13,200 Images = 13200 * 47200 = 595 MB

One Image (800 alignments) = 800*8 = 6400 Bytes

13,200 Images = 13200 * 6400 = 81 MB

Figure 3.5: The storage of high dimension feature

$$\bar{f} = \frac{1}{n} * \sum f_i \tag{3.1}$$

Step 2 High dimension image matrix:

$$\tilde{f_i} = f_i - \bar{f_i}. \tag{3.2}$$

$$A = (\tilde{f_1}, \tilde{f_2}, ..., \tilde{f_n}) \tag{3.3}$$

Step 3 covariance matrix:

$$C = AA^T, \tag{3.4}$$

The covariance matrix C is symmetric and positive definite. So the eigenvalues of C is real and non-negative.

Step 4 Eigen-decomposition:

$$Cv_i = d_i v_i, \tag{3.5}$$

d_i are the eigenvalues and v_i are the eigenvectors.

Step 5 Eigenvector matrix:

$$V = (v_1 \ v_2 \ v_3 ... v_n), \tag{3.6}$$

Step 6 Low dimension image matrix:

$$V = (v_1 \ v_2 \ v_3 ... v_j), \tag{3.7}$$

j is the dimension we want to remain.

3.2 Model Learning

As mentioned in the face model section, C_μ and C_ϵ are two unknown variables. Expectation Maximization (EM) method is used in our algorithm, which is one of the most popular machine learning approach.

3.2.1 EM

EM algorithm is an iterative method to find maximum[35][36]. In each loop, based on the objective function, a new model will be built by updating the latent variables. The loop will stop when the objective function is close to a stable maximization. We hope to set the C_μ and C_ϵ randomly and get the most appropriate joint face model by updating them automatically via EM algorithm.

We can use a simple example to describe why our algorithm needs to use EM algorithm[37]: if we want to know the height distribution of the boys and girls in our university, we cannot ask everyone so that we select 100 boys and 100 girls as the samples. Assume they follow two Gaussian models $N_1(U_1, \sigma_1), N_2(U_2, \sigma_2)$ and we do not know the U and σ. Compared to the LBP and face images, the problem of height and people seems to be similar. Because we pick each person randomly, the joint probability should be:

$$L(N) = \prod_{i=1}^{100} p(f_i; N), \tag{3.8}$$

If we want the heights of these people to be the most possible ones in our university, we need to find a N to make the L(N) biggest:

$$L(N) = \prod_{i=1}^{100} p(f_i; N), \tag{3.9}$$

If we want the heights of these people to be the most possible ones in our university, we need to find a N to make the L(N) biggest:

$$\tilde{N} = arg(Max(L(N))), \tag{3.10}$$

From this function, we can get the $N_1(\mu_1, \sigma_1)$ and $N_2(\mu_2, \sigma_2)$. It is maximum likelihood estimation (MLE), which is the basic knowledge of EM algorithm.

Now, these 200 people go together and we select one from them. We cannot know it belongs to a boys or a girls distribution and the detail parameters for each Gaussian model. So MLE cannot be used directly for the problem like this because we need to know which distribution it belongs to first.

This type of problem has two unknown variables A (boys or girls) and B (the parameter of model). The solution is to set one variable A first to get the other variable B and rectify the A by the feedback from B. For instance, we can arbitrary select one hundred people as the boys, so we can get their mean and variance

$$\tilde{N}_1 = arg(Max(L(N_1))), \tag{3.11}$$

Certainly, for the girls,

$$\tilde{N}_2 = arg(Max(L(N_2))), \tag{3.12}$$

Then, we can distribute the 200 samples to $\tilde{N}_1$ and $\tilde{N}_2$, and we can get another $\tilde{N}_1'$ and $\tilde{N}_2'$ again.

Therefore, as Fig.3.6 shown, the sample and the parameters decide which Gaussian model it should follows and the samples in each model decide the parameters.

Then, we need to use the mathematics way to prove the convergence of EM algorithm and its procedures. Assume the training set is $f_1, f_2, ..., f_m$,the samples are independent, we want to find the latent class z, make the p(f,z) be the maximum. MLE of p(f,z) is

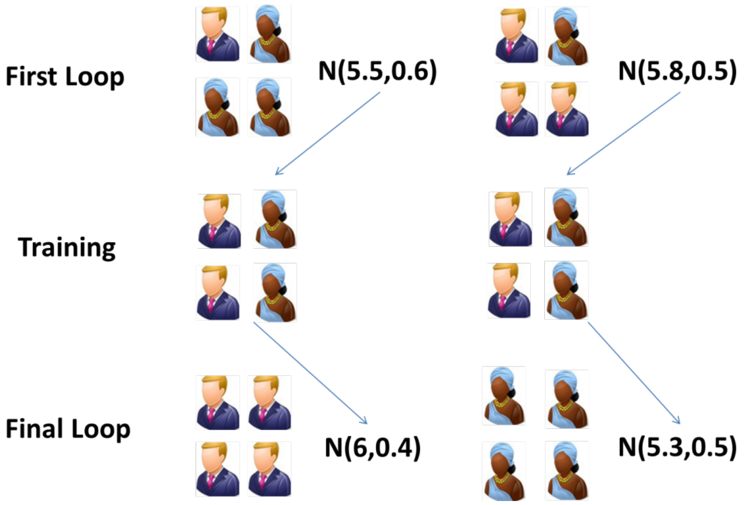


Figure 3.6: EM algorithm procedures

$$
\begin{aligned}
L(N) &= \prod_{i=1}^{m} p(f_i; N), \\
L^{'(N)} = logL(N) &= \sum_{i=1}^{m} logp(f, z; N)
\end{aligned}
\tag{3.13}
$$

EM is an effective way to solve the optimum problem with latent variables. It cannot be used to maximize the L directly but we can change the lowest boundary by E step, and optimize it by M step.

Assume Q_i represent the distribution of latent variable z, the requirement of Q_i is

$$\sum_z Q_i(z) = 1, Q_i(z) \geq 0, \tag{3.14}$$

Combine the prior equation we can get

$$\begin{aligned}\sum_{z} logp(f_i;N) &= \sum_{i} log \sum_{z^{(i)}} p(f^{(i)}, z^{(i)}; N) \\ &= \sum_{i} log \sum_{z^{(i)}} Q_i(z_{(i)}) \frac{p(f^{(i)}, z^{(i)}; N)}{Q_i(z_{(i)})} \\ &\geq \sum_{i} \sum_{z^{(i)}} Q_i(z_{(i)}) log \frac{p(f^{(i)}, z^{(i)}; N)}{Q_i(z_{(i)})}\end{aligned} \tag{3.15}$$

This function utilize the Jensen inequality (If f is convex function, X is random variable, then $E[f(X)] \geq f(EX)$).

Therefore, the procedures of the general EM algorithm are as follow:

E-step:

$$Q_i(z_{(i)}) = p(z^{(i)}|x^{(i)}; N) \tag{3.16}$$

M-step:

$$N <=> argMax \sum_{i} \sum_{z^{(i)}} Q_i(z^{(i)}) log \frac{p(f^{(i)}, z^{(i)}; N)}{Q_i(z^{(i)})} \tag{3.17}$$

If $N^{(t)}$ and $N^{(t+1)}$ are the results of n and n+1 iteration, to prove that EM algorithm is convergent, we need to prove $N^{(t)}$ and $N^{(t+1)}$ are monotonic increasing,

$$L(N^{(t)}) \leq L(N^{(t+1)}) \tag{3.18}$$

Fix $Q_i^{(t)}(z^{(i)})$, $N^{(t)}$ as a variable, take a derivative with $L(N^{(t)})$, we can get $N^{(t+1)}$ and deduce that:

$$\begin{aligned} L(N^{(t+1)}) &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) log \frac{p(f^{(i)}, z^{(i)}; N^{(t+1)})}{Q_i(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) log \frac{p(f^{(i)}, z^{(i)}; N^{(t)})}{Q_i(z^{(i)})} = L(N^{(t)}) \end{aligned} \tag{3.19}$$

3.2.2 Implementation

In our algorithm, the latent variable is a vector includes identity and variations of a face, for each human subject with m images,

$$l = [\mu; \epsilon_1; \epsilon_2; , ..., ; \epsilon_m] \tag{3.20}$$

and the input face vector is

$$f = [f_1; ...; f_m] \tag{3.21}$$

E-step: the relationship between f and l is,

$$f = Pl, P = \begin{pmatrix} I & I & 0 & ... & 0 \\ I & 0 & I & ... & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ I & 0 & 0 & ... & I \end{pmatrix}, \tag{3.22}$$

And the distributions of f and l are as follow,

$$l \sim N(0, \sigma_l), where\ \sigma_l = diag(C_\mu, C_\epsilon, ..., C_\epsilon), \tag{3.23}$$

$$f \sim N(0, \sigma_f), where\ \sigma_f = \begin{pmatrix} C_\mu + C_\epsilon & C_\mu & ... & C_\mu \\ C_\mu & C_\mu + C_\epsilon & ... & C_\mu \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C_\mu & C_\mu & ... & C_\mu + C_\epsilon \end{pmatrix}, \tag{3.24}$$

By equation 3.22-3.24, we can get the an objective function to represent the expectation of l,

$$E(l|f) = \sigma_l P^T \sigma_f^{-1} f \tag{3.25}$$

M-step: update the two covariance of C_μ and C_ϵ to change the σ_l, σ_f,

$$\begin{aligned} C_\mu &= cov(\mu), \\ C_\epsilon &= cov(\epsilon), \end{aligned} \tag{3.26}$$

Stop condition: we use a validation set and when its verification rate $r(f_1, f_2)$ does not increase any more, the algorithm will stop,

$$r_v(f_1, f_2) = log\frac{P_v(f_1, f_2|H_I)}{P_v(f_1, f_2|H_E)} \tag{3.27}$$

3.3 Computation

3.3.1 Objective Function

In the last section, we have got the objective function of expectation step of EM algorithm as follow[17][17],

$$E(l|f) = \sigma_l P^T \sigma_f^{-1} f \tag{3.28}$$

if we directly compute this equation, it will take too large memory and time to run the EM loop. Let d is the dimension of the feature and m is the number of images for each subject, we have,

$$\begin{aligned} the\ feature\ of\ \sigma_l &\sim (dm) * (dm), \\ the\ feature\ of\ \sigma_f &\sim (dm) * (dm), \\ P &\sim (dm) * (dm), \\ the\ feature\ of\ f &\sim (dm) * (dm), \end{aligned} \tag{3.29}$$

therefore, computational complexity is $O(d^3m^3)$ and memory complexity is $O(d^2m^2)$, which are too complex to compute. Now, we have use block-wise structure of the matrix to reduce the computational complexity to $O(d^3 + md^2)$ and the memory to

$O(d^2)$.

First, the inverse matrix of σ_f is,

$$\sigma_f^{-1} = \begin{pmatrix} X+Y & Y & ... & Y \\ Y & X+Y & ... & Y \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ Y & Y & ... & X+Y \end{pmatrix}, \tag{3.30}$$

From the formula,

$$\sigma_f \sigma_f^{-1} = I, \tag{3.31}$$

compute the equation of diagonal elements, for m images in a human subject,

$$(C_\mu + C_\epsilon)(X+Y) + (m-1)C_\mu Y = I, \tag{3.32}$$

For the other elements, we have,

$$(C_\mu + C_\epsilon)Y + C_\mu X + (m-1)C_\mu Y = 0, \tag{3.33}$$

left side and right side of formula 3.32 minus them of formula 3.33 separately,

$$\begin{aligned} C_\epsilon X &= I, \\ X &= C_\epsilon^{-1}, \end{aligned} \tag{3.34}$$

So, we can get Y,

$$Y = -(mC_\mu + C_\epsilon)^{-1} C_\mu C_\epsilon^{-1} \tag{3.35}$$

Put the X and Y into equation 3.30, the inverse of σ_f can be calculated. The computational complexity for X and Y are $O(d^3)$, which also results in a $O(d^3)$ computational complexity for matrix σ_f.

Take equation 3.30 ,3.34 and 3.35 into the objective function, we can get,

$$\begin{aligned} \mu &= \sum_{i=1}^{m} C_\mu (X + mY) f_i, \\ \epsilon_j &= f_j + \sum_{i=1}^{m} C_\epsilon Y f_i \end{aligned} \tag{3.36}$$

so we can see the computational complexity for computing $\mu and \epsilon$ is $O(md^2)$ and the memory complexity is $O(d^2)$.

Overall, our computational way can change the computational complexity from (d^3m^3) to $(d^3 + md^2)$ and the memory complexity from (d^2m^2) to (d^2). Generally, assume the dimension of feature is 1000 after the PCA step and the number of images for one human subject is m = 20, the total reduction is,

$$\begin{aligned} Computational\ Complexity: \frac{1000^3 * 20^3}{1000^3 + 20 * 1000^2} &= 8000, \\ Memory\ Complexity: \frac{1000^2 * 20^2}{1000^2} &= 400, \end{aligned} \tag{3.37}$$

the more images in a subject, the more computational reduction we can get.

3.3.2 Log Likelihood Ratio

From the formula 3.27,

$$r(f_1, f_2) = log\frac{P(f_1, f_2|H_I)}{P(f_1, f_2|H_E)} \tag{3.38}$$

To compute it easier, we use a closed form after simple algebra operations:

$$r(f_1, f_2) = f_1^T A f_1 + f_2^T A f_2 - 2 f_1^T Y f_2^T, \tag{3.39}$$

where

$$A = (C_\mu + C_\epsilon)^{-1} - (X + Y), \tag{3.40}$$

$$\begin{pmatrix} X+Y & Y \\ Y & X+Y \end{pmatrix} = \begin{pmatrix} C_\mu + C_\epsilon & C_\mu \\ C_\mu & C_\mu + C_\epsilon \end{pmatrix}^{-1} \tag{3.41}$$

As m = 2 for a pair, put formula 3.34 and 3.35 into above equation,

$$Y = -(2 * C_\mu + C_\epsilon)^{-1} C_\mu C_\epsilon^{-1} \tag{3.42}$$

$$A = (C_\mu + C_\epsilon)^{-1} - (C_\epsilon^{-1} - (2 * C_\mu + C_\epsilon)^{-1} C_\mu C_\epsilon^{-1}) \tag{3.43}$$

Therefore, we can use only C_μ and C_ϵ to represent log likelihood ratio.

Chapter 4

Experiment

4.1 Dataset

Labeled Faces in the Wild (*LFW*) is established by the computer vision lab of University of Massachusetts. There are totally 13233 images and 5749 human subjects. It is a good and difficult database for face recognition but it is not appropriate to train because there are not enough face images for each human subject. Though only 96 people have more than 15 images for each subject, the training process of face verification algorithm does not need a wide and deep database.

In our experiment, the images in LFW are just directly divided by the number of images for each subject, I set two types approaches for training:

1. If this human subject includes more than m images, we choose m images to train.

2. If this human subject includes more than m images, we choose all images in

this subject to train.

Obviously, the more images we train, the more information we can have. However, there may be over-fitting when we use EM algorithm and the result may not be better with the more images for each subject. For different number of images,

LFW	
m	n
1	5749
2	1680
3	901
4	610
5	423
6	311
7	256
8	217
9	184
10	158
11	143
12	127
13	117
14	106
15	96

Table 4.1: The details of LFW

The Table 4.1 shows that there will be n subjects that contains more than m images. As mentioned above, the size of all images is 250 * 250 and the face in each image have complex background.

To test our face verification algorithm, we select 3000 pairs intra-subject images and 3000 pairs extra-subject images. Intra-subject images represent the two images in a pair belong to the same subject and extra-subject images represent the opposition. From this 6000 pairs images, we pick 300 pairs as the validation set in the loop of EM algorithm.

4.2 Results

There are many adjustable parameters in our method. In this section, we will change them to see the influence for each variable from the various results. Because we hope to find the best parameters to implement our algorithm and prove it is a robust and competitive approach.

4.2.1 Initialization

In our algorithm, we can mainly control 5 parameters:

1. The minimum number of images in each subject m: as the data in Table 4.1 shown, it decides the number of subjects.

2. The training type k: we have talked in last section about it. Combined with m, they can decide the totally number of images.

3. The dimension of feature d: we use PCA to reduce dimension of LBP from 5900 to a low dimension and keep the verification rate as high as possible. It decides how many information we need and how long we need to train.

4. C_μ and C_ϵ: to represent our face model, we just use only these two variables. Because EM algorithm will learn the optimization automatically, we just use two random positive definite matrix as initial ones.

Besides, to balance the running time and verification rate, we use a validation set[38] in EM algorithm[38], the detail process as follow,

Fig.4.1 indicates that if the verification rate do not increase any more within 6 loops, we will choose the result of first loop in these 6 loops as the final verification rate, which is the best result.

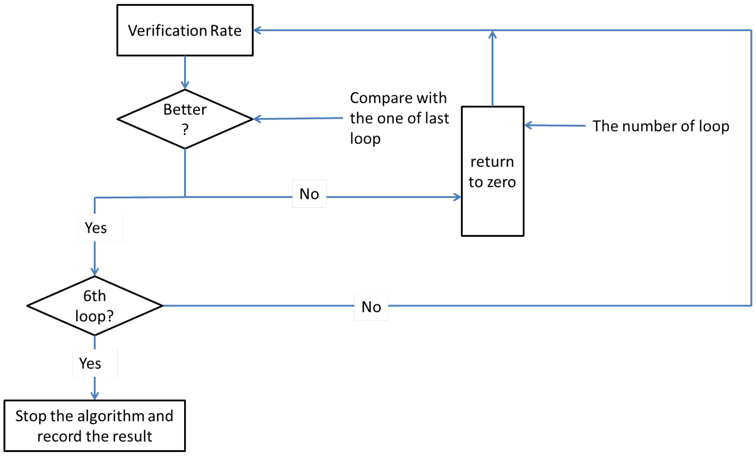


Figure 4.1: Flow chart of validating process

4.2.2 Implementation

First, we will show how the dimensions of LBP affect our algorithm, the number of images for each subject m=4 and the training type=1,

Dimension	Intra-result	Extra-result	Final Result
100	0.892222	0.550741	0.721481
200	0.774444	0.799630	0.787037
300	0.802963	0.832593	0.817778
500	0.551852	0.986296	0.769074
800	0.295926	1.000000	0.647963

Table 4.2: type 1 with different dimensions

In table 4.2, intra-result represents the verification rate of the pairs that whose images belong to the same human subject. When the dimensions are 100, it reaches nearly 90 percent and goes down to nearly 30 percent with increasing dimension to 800. On the contrary, extra-result shows a upward trend from 55 percent to 1. It is reasonable that the number of intra-pairs is equal to the number of extra-pairs. The final result performs better and better before the dimension is more than 400 and the best one reaches nearly 82 percent.

Overall, with the dimension of feature increasing, the extra-result changes faster if the dimension is low enough and the intra-result changes faster when the dimensions go high. For m=4, type=1, verification rate reaches best at 82 percent when the dimensions are 300.

Obviously, comparing the two training types, type 1 refer to less images than type 2. However, if we just change the dimensions of feature and remain the number of images for each subject m=4, type 2 shows a much different trend with type 1,

Dimension	Intra-result	Extra-result	Final Result
100	0.949259	0.464815	0.707037
200	0.975185	0.455926	0.715556
300	0.965185	0.543333	0.754259
500	0.904815	0.835556	0.870185
800	0.852593	0.938148	0.895370

Table 4.3: type 2 with different dimensions

When training type 2 is used, we can see the performance is much better than type 1. All the results are more than 85 percent when the dimensions are 800. On the one side, no matter how many dimensions of the feature, the intra-result is stable and excellent. On the other side, the extra-result shows a much improvement with the increasing dimensions. We can infer that: when we use type 2 to train the feature with low dimension, our algorithm cannot distinguish the two people in extra-pairs but the high dimensions feature will offer much more useful information than redundancy.

Then, we do another experiment to compare the detail of difference between type 1 and type 2.

As shown as Fig.4.2, all the intra-results of type 2 perform better than type 1, which illustrates that our LBP feature is good enough to catch the "similar information" for same person from each face image. Meanwhile, we need to check if our

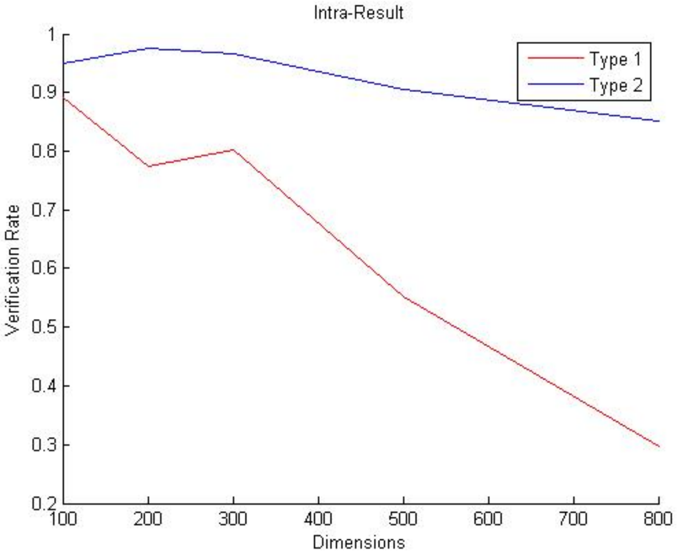


Figure 4.2: intra-result with different type

feature can also catch the "distinctive information",

Although both two curves show a upward trend, we can see that extra-result of type 1 is always better than the one of type 2 from Fig.4.3, but the difference between them are not larger than the intra-result. It is easy to understand that our algorithm will receive more information when the dimensions go up. Because there is more opportunity to get "distinctive information" due to the principle of PCA. And the EM algorithm is used to balance the two type information.

In Fig.4.4, the performance of type 2 is worse than type 1 in the beginning and when the dimensions of feature reach 400, the performance of type 2 is better and better, which shows nearly state of the art with 800 dimensions of feature.

Dimension	Intra-result	Extra-result	Final Result
1000	0.863704	0.942963	0.903333
1500	0.641724	0.843451	0.742588

Table 4.4: More than 1000 dimensions

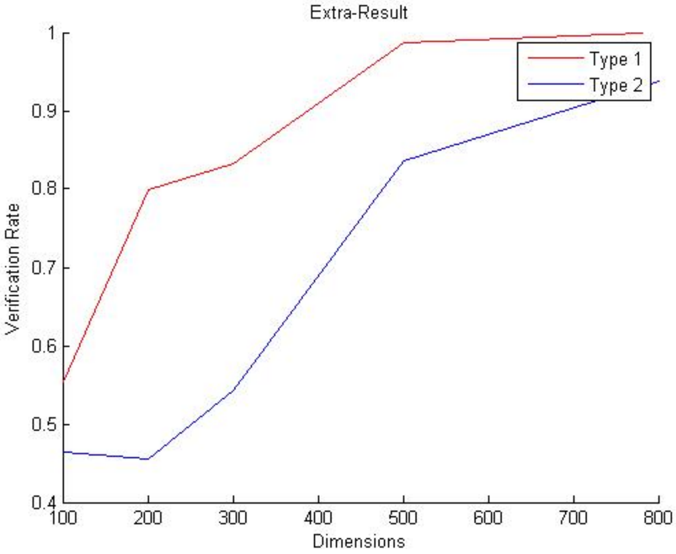


Figure 4.3: extra-result with different type

If the number of images for each subject m=4, the final-result of our algorithm performs best when the type = 2 and the dimensions of feature d=1000. The table.4.4 shows that all the results do not become better, which may be due to a over-fitting training.

Finally, to get a convincing conclusion, we also need to compare different number of images for each subject. We fix the variables type = 2 and d = 100 and change m to 1 and 10, the results of our algorithm are shown as below,

m	subjects	Intra-result	Extra-result	Final Result
2	1680	0.844074	0.562222	0.703149
4	610	0.949259	0.464815	0.707037
10	158	0.958519	0.367778	0.663149

Table 4.5: Different m variable(d=100)

The table.4.5 tells that the intra-result performs better while the extra-result get worse and the final-results are similar. It may be result from the too low dimensions

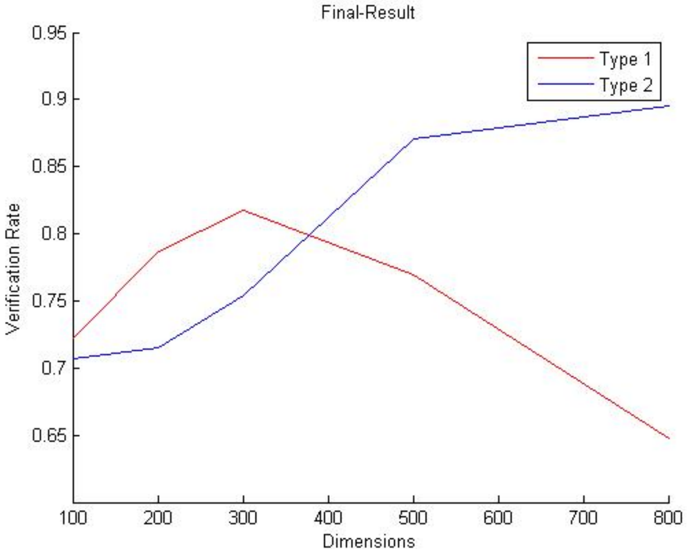


Figure 4.4: final-result with different type

of feature, the difference is not outstanding enough. So we increase the dimensions,

m	subjects	Intra-result	Extra-result	Final Result
2	1680	0.923333	0.735926	0.829631
4	610	0.925111	0.744453	0.827782
10	158	0.842593	0.810370	0.826482

Table 4.6: Different m variable(d=400)

As Fig.4.6, all the results are nearly 82% and more images result in higher verification rate. Actually, our algorithm with 400 dimensions has showed us a relative reasonable trend. Therefore, we continue to change a larger d=800 to see the variation,

It is amazing that the high dimensions result in a big improvement for the final performance, which has been over 92 %. We can see that the intra-result has a much larger change with the increasing subjects. To compare the data from table.4.7 and table.4.6, we can infer that: intra-result is influenced by the dimensions of feature

m	subjects	Intra-result	Extra-result	Final Result
2	1680	0.905926	0.948519	0.927223
4	610	0.852593	0.938148	0.895370
10	158	0.349630	0.998148	0.673890

Table 4.7: Different m variable(d=800)

and more training images results in the more effects; extra-result is controlled by the number of subjects or total number of images.

After the experiments of changing all the parameters in our algorithm, to show the excellent performance of our method, we compare our Joint Bayesian method with the other algorithms by using same database,

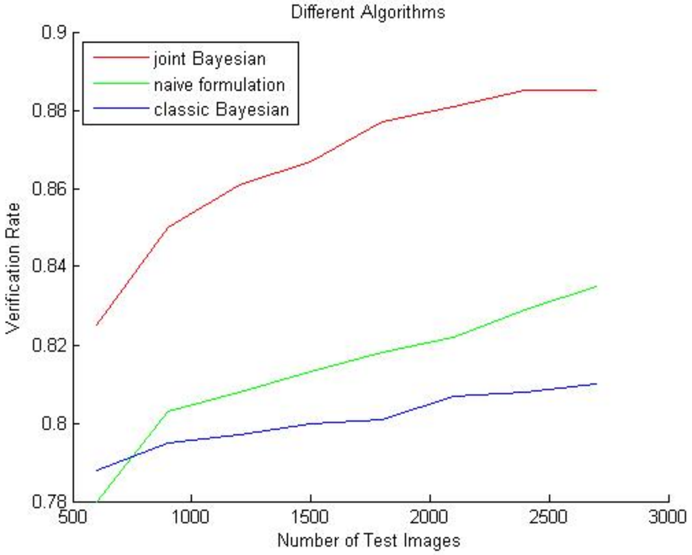


Figure 4.5: Comparison with other Bayesian methods

From the curves of the Fig.4.5, we can see that our new joint Bayesian method performs much better than the traditional Bayesian ways and the verification rate goes up with the more number of test pairs.

In conclusion, from so many experiments, we have analyze all the parameters and find their impacts for the whole algorithm. Indeed, there are some cross datasets among training images, validation pairs and test pairs. To avoid training the people that may appear in our test set, we do not set the number of subjects is equal to 5749. And the best performance of our algorithm can be over 93 % with the parameters m=2, type=2 and d=1200.

Chapter 5

Summary of Contributions and Future Work

5.1 Contributions

We have introduced a novel face model and showed the superiority by comparing with the other algorithms. To balance the variables of our algorithm and the final verification rate, we implement almost every conditions and get the best result more than 93 % when m=2, d=1200 and type=2. However, based on the restriction of hardware, we have not test the dimensions over 2000 because the memory of our sever is not enough and the training time is more than one month.

We beat most good algorithms and the contribution of my project is mainly about:

1. Implement various experiments for this novel algorithm and analyze the function of every parameters in our method.

2. Add many new elements to improve the original algorithm. For example, we

differentiate the type 1 and type 2 for joint Bayesian method and add validation set as stop condition. The final result of our improved algorithm has been over 93 %.

5.2 Future Works

We prepare to use this new face model into our face verification system, which includes 3 completely algorithms (detection, alignment, verification). As shown as Fig.5.1, our interface is the feature of the face image, which is a 6000 dimensions vector.

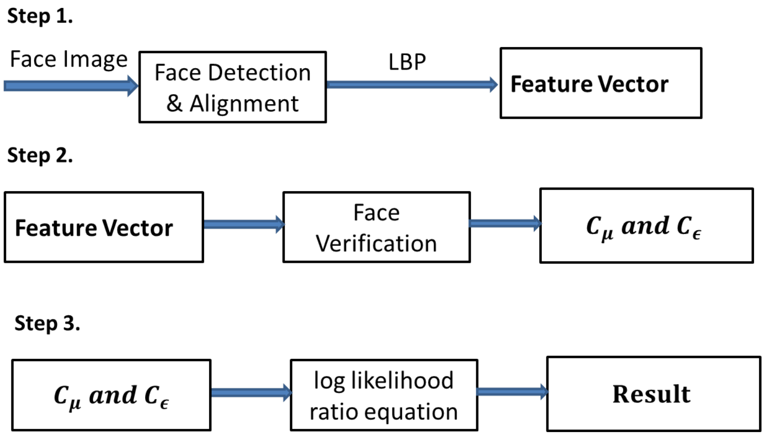


Figure 5.1: future work

Our face detection method is improved from the implementation of OpenCV[39]. It will output a rectangle region as the input of face alignment method. Indeed, we just use these regions as the new cropped face images and improve the algorithm of [16] to get alignment points. Finally, all the other procedures are the same as the face verification method has been mentioned above.

Bibliography

[1] Cees G.M. Snoek Efstratios Gavves and Arnold W.M. Smeulders. Convex reduction of high-dimensional kernels for visual classification. *CVPR*, 2012.

[2] Fang Wen Jian Sun Dong Chen, Xudong Cao. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. *CVPR*, pages 3025–3032, 2013.

[3] Q.M. Jonathan Wu M.A. Sid-Ahmed A.A. Mohammed, R. Minhas. Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition*, 32:2588–2597, 2011.

[4] Pinto N. Cox, D. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. *Automatic Face & Gesture Recognition and Workshops*, pages 8–15, 2011.

[5] A.J.Mistlin A.O'Toole, Mistlin and A.J.Chitty. A physical system approach to recognition memory for spatially transformed faces. *Neural Networks*, 1:179–199, 1988.

[6] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[7] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America*, 14:1724–1733, 1997.

[8] I. Craw and P.Camron. Face recognition by computer. *British Machine Vision Conference*, pages 498–507, 1992.

[9] Scale-invariant feature transform. *Wikipedia*.

[10] Histogram of oriented gradients. *Wikipedia*.

[11] Gabor filter. *Wikipedia*.

[12] Abdenour Hadid Timo Ahonen and Matti Pietik. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence*, 28:2037–2041, 2006.

[13] Yin Q. Tang X. Sun J Cao, Z. Face recognition with learning-based descriptor. *CVPR*, 2010.

[14] Honglak Lee Erik Learned-Miller Gary B. Huang, Marwan Mattar. Learning to align from scratch. *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[15] Michael Jones Paul Viola. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.

[16] Xiao R. Wen F. Sun J Liang, L. Face alignment via component-based discriminative search. *ECCV*, 2008.

[17] L. Wang F. Wen D. Chen, X. Cao and J. Sun. Bayesian face revisited: A joint formulation. *ECCV*, pages 566–579, 2012.

[18] Jeffrey Ho Kuang-Chih Lee and David Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 27, 2005.

[19] T. Berg E. Learned-Miller G. B. Huang, M. Ramesh and A. Hanson. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2008.

[20] Jebara T. Pentland Moghaddam, B. Bayesian face recognition. *Pattern Recognition*, 33:1771–1782, 2000.

[21] Tang X Wang, X. Bayesian face recognition using gabor features. pages 70–73, 2003.

[22] Tang X Wang, X. A unified framework for subspace face recognition. pages 1222–1228, 2004.

[23] Tang X Li, Z. Bayesian face recognition using support vector machine and face clustering. *CVPR*, pages 1222–1228, 2004.

[24] J. Kittler C. Chan and K. Messer. Multi-scale local binary pattern histograms for face recognition. *Advances in biometrics*, pages 809–818, 2007.

[25] Xiaoou Tang Yi Sun, Xiaogang Wang. Hybrid deep learning for face verification. *ICCV*, 2013.

[26] Hespanha J.P. Kriegman D.J Belhumeur, P.N. Eigenfaces vs. fisherface:srecognition using class specific linear projection. *PAMI*, 19:711–720, 1997.

[27] S Ioffe. Probabilistic linear discriminant analysis. *ECCV*, 2006.

[28] Li P. Fu Y. Mohammed U. Elder-J Prince, S. Probabilistic models for inference about identity. *PAMI*, 34:144–157, 2012.

[29] Multivariate normal distribution. *Wikipedia*.

[30] Pietikainen M. Maenpaa T Ojala, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24:971–987, 2002.

[31] Shiguang Shan Wen Li Xilin Chen Meina Kan, Dong Xu. Learning prototype hyperplanes for face verification in the wild. *Image Processing*, 22:3310–3316, 2013.

[32] Wu Kaining and Cao Hanqiang. Face recognition via sparse representation and local binary patterns. *International Conference on Advanced Computer Theory and Engineering*, 2012.

[33] Wu Kaining and Cao Hanqiang. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *ICPR*, 1:582–585, 1994.

[34] Principal component analysis. *Wikipedia*.

[35] A. Wright J. Wenli Xu Yi Ma Yigang Peng, Ganesh. Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence*, 34:2233–2246, 2012.

[36] Bao-Gang Hu Ran He, Wei-Shi Zheng. Maximum correntropy criterion for robust face recognition. *Pattern Analysis and Machine Intelligence*, 34:1561–1576, 2011.

[37] Geoffrey J McLachlan. *The EM algorithm and extensions.*

[38] Robert Sabourin Jean-Franois Connolly, Eric GrangerCorresponding. An adaptive classification system for video-based face recognition. *Information Sciences*, 192:50–70, 2012.

[39] Jones M. Viola, P. Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:511–518, 2001.