LOCAL BINARY PATTERN NETWORK : A DEEP LEARNING APPROACH FOR FACE RECOGNITION

by

MENG XI

B.Eng., Beijing University of Chemical Technology, 2005.
M.Eng., Peking University, 2008.

THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
MATHEMATICAL, COMPUTER, AND PHYSICAL SCIENCES
(COMPUTER SCIENCE)

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

2015

Abstract

Deep learning is well known as a method to extract hierarchical representations of data. This method has been widely implemented in many fields, including image classification, speech recognition, natural language processing, etc. Over the past decade, deep learning has made a great progress in solving face recognition problems due to its effectiveness. In this thesis a novel deep learning multilayer hierarchy based methodology, named Local Binary Pattern Network (LBPNet), is proposed. Unlike the shallow LBP method, LBPNet performs multi-scale analysis and gains high-level representations from low-level overlapped features in a systematic manner.

The LBPNet deep learning network is generated by retaining the topology of Convolutional Neural Network (CNN) and replacing its trainable kernel with the off-the-shelf computer vision descriptor, the LBP descriptor. This enables LBPNet to achieve a high recognition accuracy without requiring costly model learning approach on massive data. LBPNet progressively extracts features from input images from test and training data through multiple processing layers, pairwisely measures the similarity of extracted features in regional level, and then performs the classification based on the aggregated similarity values.

Through extensive numerical experiments using the popular benchmarks (i.e., FERET, LFW and YTF), LBPNet has shown the promising results. Its results outperform (on FERET) or are comparable (on LFW and FERET) to other methods in the same categories, which are single descriptor based unsupervised learning methods on FERET and LFW, and single descriptor based supervised learning methods with image-restricted no outside data settings on LFW and YTF, respectively.

Contents

List of Figures

Acknowledgements

First, I would like to express my sincere appreciation to my supervisor, Prof. Liang Chen, for his immense and continuous support throughout my studies and life. His guidance always inspire and encourage me in all the time of the research and writing this thesis. Without his patient and motivated guidance, and persistent help, this thesis is impossible.

I would also like to thank Prof. Jernej Polajnar who introduced me to the distributed system, and Prof. Youmin Tang who guided me in the statistics.

In addition, I would like to thank to my colleagues, Negar Hassanpour, Yunke Li, Tony Zhuang for their inspiring suggestions and discussions during the work of the research.

Thanks to my friends Ben Ng, Chuyi Wang, Zarina Hasanov in Prince George, who provided their warmest help and made my life so colourful and fun.

Last but not least, I would like to thank my family, my wife Jing Liang and my son Zhihong Xi. Their love and encouragement is one of the most supportive strength for me.

Chapter 1

Introduction

Face recognition is a sub-division of image processing, computer vision, pattern recognition, and machining learning. The application of face recognition systems is using computer algorithms to identify/verify human faces in still images or video clips. It continuously attracts interests from researchers because of its wide range applications in the real world, such as security, computer entertainment, multimedia management, law enforcement and surveillance [6, 7].

Since there exist many mystical parts in the perception of human faces, face recognition systems are built using statistical models with only a little prior knowledge. In such systems, the images/videos are represented as one or a set of numerical metrics. The recognition system itself is a function that accepts matrices as inputs of images and returns their similarities. For instance, to find out how much two faces look like to each other, a distance measure can be employed to compute the difference between the two corresponding matrices. Such difference then reflects how similar these two faces are.

1.1 Challenges of Face Recognition

In the past decade many methods have been proposed to improve the accuracy of face recognition significantly. However, there remains a lot of challenges. Figure 1.1 shows a set of sample pictures that the same person looks dramatically unlike due to different photo-taking environments. It is even challenging for a human being to identify these faces as one person from these pictures.

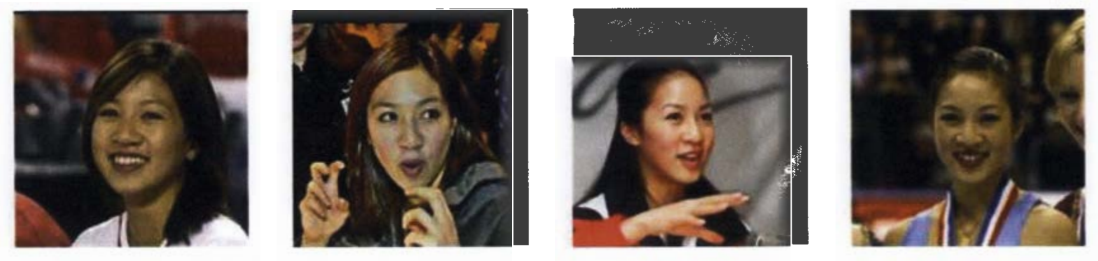


Figure 1.1: The same person looks dramatically different due to pictures taken from different pose angle, makeups, lightning condition, aging, etc. The samples are from LFW dataset [1, 2].

Some of many variance of human faces which bring a lot of uncertainties and significantly affect the recognition accuracy are emphasized as follows:

- Illumination, as one of the well-studied unstable factors, is brought by the change of lighting condition which has a non-linear influence on the image even when holding other conditions unchanged. Although illumination problem has been largely solved in certain conditions [1], developing illumination-robust algorithm for more difficult lighting condition is still an ongoing research since the problem becomes more complicated when combining other effects with illumination.

- Pose angle is another major reason of uncertainties. Since the image of human face is indeed obtained by a projecting 3-dimensional object into a 2-dimensional

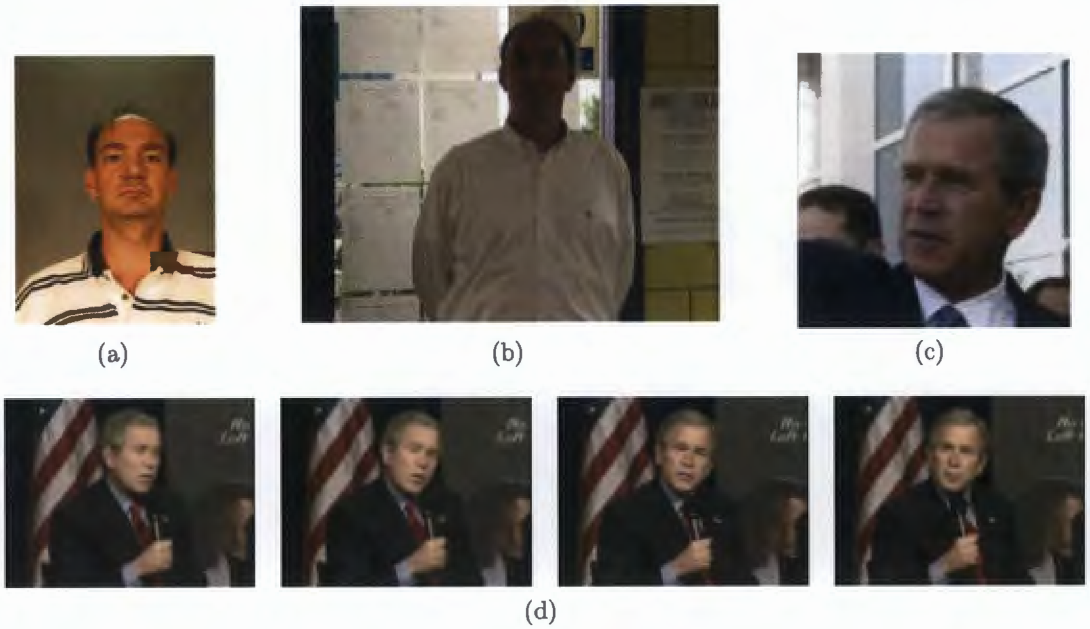
[1] For the Fc probe set of the FERET benchmark which aims at different illumination, current state-of-the-art method has achieved 100% recognition accuracy of it

plane, the projected image is inevitably distorted. The size and shape of organs in face (e.g., eyes, nose) change in different shooting angles because of effect of perspective. Additionally, important information may be lost since large area of face is shrunk into a small region in picture, or even occluded. Furthermore, it also brings unpredictable effects on illumination.

- Facial expression brings uncertainty due to the distortion of the face itself: organs leave their original place and change their shapes; even the 3-dimensional shape of the face changes because of the muscle movement.

Other difficulties in face recognition include occlusion, aging, makeups, image quality, etc.

Admittedly, high recognition accuracy has been achieved on some benchmark datasets. For example, the error rate of Eighenface is reported as low as 7.3% in Yale data set [8]. However, recent researchers' interests have begun to focus on more challenging tasks, including pictures taken in uncontrolled environment (e.g., Face Recognition Grand Challenge [3]), in unconstrained environment (e.g., Labeled Faces in the Wild [1, 2]) and video-based face recognition (e.g., YouTube Faces [4]) which are shown in Figure 1.2. The images or videos in these datasets are taken with wide variation in illumination, expression, pose angle, even the picture quality. Recognition for these datasets is considered to be more challenging than in controlled environment, but the algorithms applicable for them are also more practical in real world face recognition tasks.



(a) (b) (c)

(d)

Figure 1.2: Some example pictures from different types of datasets. (a) and (b) are from FRGC [3], where (a) was taken in controlled environment and (b) was in uncontrolled environment; (c) is unconstrained image from LFW [1, 2]; (d) are some sample frames from a unconstrained video clip in YTF [4].

1.2 Overview of this thesis

Recently deep learning has brought a lot of attentions because of the state-of-the-art results it achieved in image classification tasks [9, 10, 11, 12, 13, 14, 15, 16, 17]. Deep learning is one brunch of machine learning which tends to extract high level abstractions or representations of data through multiple processing layers [18]. A hierarchical architecture can be formed through sequential connections between multiple layers. The latter layers in the hierarchy extract higher level of abstractions from the lower level ones extracted by the earlier layers. In addition, features are extracted in a heavily overlapped manner. This means that a low-level feature can contribute to multiple high-level features in the later layer.

Convolutional Neural Network (CNN) is one of the most commonly studied deep learning architectures, which can be viewed as a variant of multilayer perceptron (MLP) neural network. CNN obtains the facial discriminative representations from a set of hierarchically connected and trainable convolutional kernels [5, 18]. Comparing with other regular face recognition methods, training CNN is troublesome. Difficulties in CNN are generally twofold: (i) the learning approach itself is computation expensive due to a large amount of parameters in sequentially connected multiple layers, which makes the convergence undesirably time-consuming; (ii) overfitting is more likely to occur due to the existence of thousands of parameters in this model. The former issue is primarily solved using powerful computers and leveraging hardware accelerating techniques (e.g., GPU computing). To tackle the latter issue, in the case of face recognition, many state-of-the-art systems leverage massive external data to learn their networks [9, 10, 11, 12, 13, 14]. However, we believe that these are just workarounds by utilizing more computing resource rather than final solutions. Considering the complexities of CNN are mainly attributed to its trainable kernels, the question we want to address here is the possibility to replace the convolutional kernels with *off-the-shelf* computer vision descriptors such that the framework is capable for the high-level feature extraction on dense data with only a few of adjustable parameters. This can help avoid the costly training process and therefore reducing the need of training data.

In this thesis, a deep network based on LBP descriptor is proposed, which is named as Local Binary Pattern Network (LBPNet). Two filters are used in LBPNet, which are based on Local Binary Pattern (LBP) and Principle Component Analysis (PCA) techniques, respectively. The over-complete patch-based features are extracted hierarchically by these two filters. After feature extraction, the LBPNet employs a simple network to measure the similarity of the extracted features. Major characteristics of the proposed LBPNet are summarized in the following:

Feature extraction in dense grid: Both of the two filters are replicated densely in layers.

Multilayer architecture: The representations are extracted hierarchically: the latter layer extracts a higher level of abstractions from the lower level ones of the earlier layer.

Partially connected layer: Filters only compute based on the selected subset of the inputs from the earlier layer.

Multi-scale analysis: Filters with different parameters are used in each of the layers to capture multi-scale statistics.

Unsupervised learning: Since both LBP and PCA are unsupervised learning algorithms, LBNet is capable to perform unsupervised learning on data.

Since LBPNet contains all the fundamental characteristics of deep learning architecture, it can be classified as a simplified deep network with *hand-craft* filters. Comparing with the regular CNN architectures, LBPNet retains the key CNN architectural features but simplifies the model by replacing its trainable kernel with the off-the-shelf computer vision descriptor, the LBP descriptor, to avoid the costly training approach. The framework proposed in this thesis significantly outperforms the original LBP approach.

1.3 Contributions

The main contributions of this thesis are summarized as follows:

This thesis presents a novel deep learning based methodology for face recognition named Local Binary Pattern Network (LBPNet). It extracts and compares high-level over-complete facial descriptors hierarchically based on a single-type LBP descriptor. By borrowing the deep network architecture from Convolutional Neural Network while replacing its trainable kernel to off-the-shelf computer vision descriptors, LBPNet is able to perform multi-scale analysis on dense features hierarchically while only requiring a simple training approach on a relatively small training set. In addition, the LBPNet is capable for both supervised and unsupervised learning algorithm.

By embedding into our framework the original LBP approach performance boosts significantly. Experimental results on several public benchmarks (i.e., FERET, LFW, YTF) have shown that the LBPNet outperforms or is comparable to other single descriptor based methods under the same protocols, including unsupervised learning protocol on FERET and LFW and image-restricted no outside data protocol on LFW and YTF, respectively.

1.4 Organization of this thesis

The rest of this document is organized as follows:

Chapter 2 introduces the general face recognition pipeline as well as several baseline methods of it, namely, LBP, subspace projection (PCA and LDA) and classifiers including Nearest Neighbourhood classifiers and Support Vector Machine.

In Chapter 3, several state-of-the-art algorithms which are related to our works or inspired us are introduced. In the first section, several over-complete feature extraction algorithm are introduced. Next in the second chapter, the patch-based systems for face recognition are discussed. Finally, the third section introduces the architecture of Convolutional Neural Network.

Chapter 4 elaborates the proposed baseline LBP and LBPNet methods. The detail designs of each layers of LBPNet are given in the first section. Next, the scheme for video based face recognition is introduced.

Chapter 5 includes the introduction of the benchmarks employed in the experiment (i.e., FERET, LFW, YTF) as well as the parameter settings for each dataset.

Chapter 6 reports the experimental results of LBPNet. Its results outperform (on FERET) or are comparable (on LFW and FERET) to other methods in the same categories, which are single descriptor based unsupervised learning methods on FERET and LFW, and single descriptor based supervised learning methods with image-restricted no outside data settings on LFW and YTF, respectively. Additionally, results from the baseline LBP methods of LBPNet are also reported to demonstrate that the deep learning architecture of LBPNet improves the performance fundamentally.

In the end, Chapter 7 summarizes this thesis and suggests some future directions.

Chapter 2

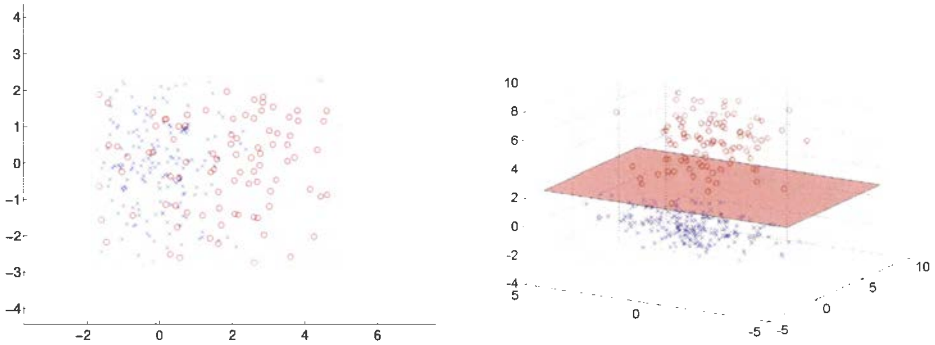
Background

There exist two types of face recognition tasks: face verification and face identification. The identification systems find the identities of unknown faces according to the known faces, whereas the verification systems confirm or reject two faces having the same identity. The dataset of known faces is called gallery set while the set of unknown faces is probe set. The general processing pipeline of the face recognition system has several important stages as follows.

Face detection: The first stage of face recognition is face detection. It finds the facial area in image or video frame and passes it to the next stage.

Face normalization: Face normalization module performs preparation for the following stages. It contains two components: geometric normalization component which rotates and scales the face to the same position among all images; photometric normalization component which performs illumination adjustments.

Feature extraction: A feature is a numerical representation of image. It is either directly computed from intensity image or from other features of this image. Extracted features are robust to variances and easy to classify compared to intensity images. In mathematical context, feature extraction is a projection from input space into feature space (Figure 2.1).



(a) (b)

Figure 2.1: Samples are projected from the input space (a) into feature space (b). Samples are hard to be separated in input space, but they are linearly-separable in feature space.

The features can roughly be divided into two categories: low level features (e.g., LBP [19], SIFT [20], Gabor [21]) and high level features which are computed from low level ones. The high level features are more informative and more robust of variances. Section 2.1 will have a briefly introduction on the low level descriptor LBP and Chapter 3 will discuss several high level feature extraction frameworks.

Dimensionality reduction: One common problem of face recognition methods is that the extracted features are of high-dimensionality. Therefore, techniques of dimensionality reduction are highly desired. It is a critical step in many state-of-the-art methods [22, 23, 24, 25, 26]. Linear subspace projection, as one of the widely used techniques, is introduced in 2.2

Classification: The last stage of the face recognition pipeline is to classify the faces through the extracted features. The classifier evaluates the similarity level of the faces and makes decision according to it. Several classifiers are discussed in Section 2.3.

Note that not every work includes every stage mentioned above. Particular works usually just focus on improving one or several stages while leaving other stages as it is by leveraging the existing algorithms or results. Also in some works, feature extraction or/and dimensionality reduction stages are absent.

2.1 Local Binary Pattern

The Local Binary Pattern (LBP) operator, introduced by Ojala *et al.* [27], is a regional descriptor-based approach for texture description. It was latter introduced into face recognition area by Ahonen *et al.* [19].

The LBP generation approach in [19] is described as follows. To start with, the LBP map is built by applying LBP operator. For each pixel of the image, the operator thresholds its surrounding 3×3 pixels: for each neighbourhood pixel, if its grey scale value is greater than the centre pixel, we assign a binary number 1 to it ; otherwise, we assign a 0 to it. Afterwards, all the binary numbers are stacked into one vector as the label of the centre pixel. The diagram of encoding scheme is shown in Figure 2.2, the centre pixel is labelled as 01011010 in binary or 90 in decimal.

One extension of this basic operator is to allow neighbourhoods of arbitrary size and numbers as shown in Figure 2.3. The notation $LBP_{P,R}$ are used to denote a LBP operator in which P points are sampled on a circle of radius of R. When the sampling

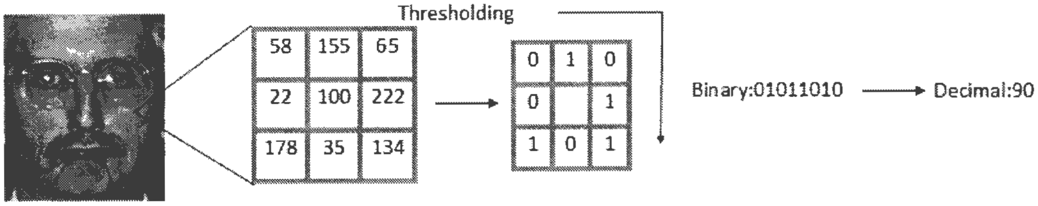


Figure 2.2: The basic LBP operator

point is not in the centre of pixel, the bilinear interpolation will be used to obtain its value.

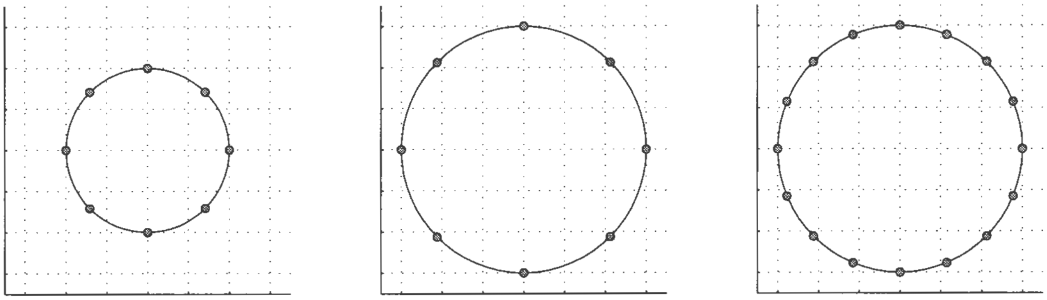


Figure 2.3: LBP operators which can be denoted as $LBP_{8,2}$, $LBP_{8,3}$, $LBP_{16,3}$

Another extension is the uniform pattern which is denoted as LBP^{u2}. A LBP label is called uniform when at most two bitwise transition from 0 to 1 or vise versa is contained when considering it as a circular. Some examples are shown in Figure 2.4. For the operator of 8 sampling points, there exist 58 uniform patterns (Figure 2.5). According to [19], around 90% pattern in $LBP_{8,1}$ is uniformed. LBP labels in this case can be further encoded into 59 numbers: one for non-uniform pattern and others for uniform pattern.

The second step of [19] is to generate LBP histogram features. The LBP image is divided into several non-overlapped cells, and the histograms are computed in each

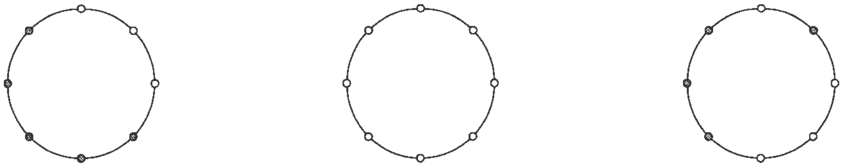


Figure 2.4: Diagrams represent LBP label "00011111", "00000000" and "10001110" respectively. The leftmost two are uniform patterns while the rightmost one is not as it contains more than two bitwise transition.

cell which is defined as histogram function H:

$$H_i = \sum B((LBP_{P,R}^{u2}(x, y)) = i)|i \in [0, n] \tag{2.1}$$

where i denotes different encoded LBP labels, (x, y) are coordinates of circle centre, and

$$B(v) = \begin{cases} 1, \text{when } v \text{ is true} \\ 0, \text{when } v \text{ is false} \end{cases} \tag{2.2}$$

This histogram function counts the number of different LBP labels in a specific area and then stack the result into one string. The overall LBP histogram descriptor of this image is the concatenation of the histograms of all cells. The diagram of the whole approach is shown in Figure 2.6.

2.2 Linear Subspace Projection

Linear subspace projection seeks a transformation matrix W to project the input vector p into a lower dimensionality space expressed as

$$p\prime = W^T p \tag{2.3}$$

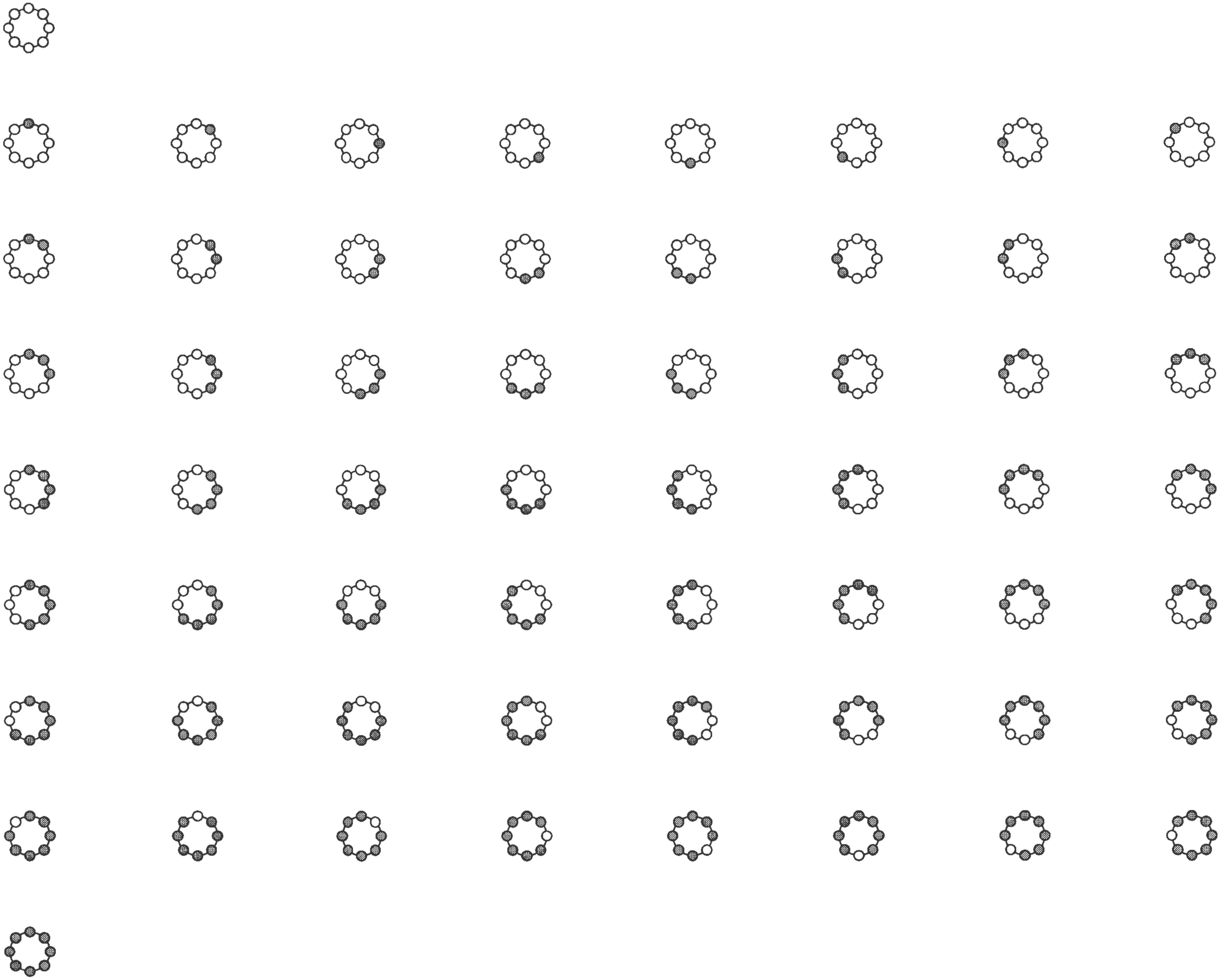


Figure 2.5: 58 uniform pattern of $LBP^{u2}_{8,R}$. From top to bottom, the number of 1 in the LBP label increases; from left to right, the label rotates as a circular.

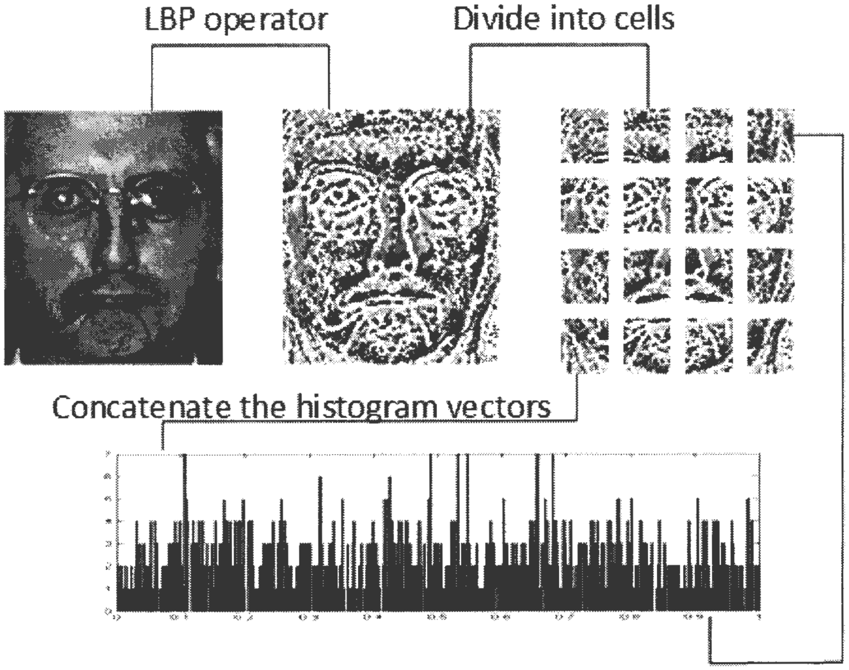


Figure 2.6: A schematic diagram of LBP descriptor extraction pipeline

If $p \in \mathbb{R}^m$, $W \in \mathbb{R}^{m \times n}$, $m > n$, then $p\prime \in \mathbb{R}^{n \times 1}$ is narrower than p. Additionally, subspace projection can also be employed as feature extractor [8]. In the rest of this section, two linear projection methods, Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA), will be introduced.

2.2.1 Principle Component Analysis

PCA projection is a orthogonal projection. If the input data are correlated, it is possible to generate output data of lower dimension than the input data while keeping as much as possible variability of the input data. By projecting into PCA subspace, the variables in the input are decorrelated to each other. Figure 2.7 shows an example of PCA the input data contains 2 correlated attributes, and PCA finds out an optimized direction of projection to reduce the dimension of the data to 1. This direction keeps most of the variability of the input data. PCA is useful in the context of face recognition due to its high dimensional data and limited number of samples. Reducing the dimensionality of data helps avoid overfitting and consequently improve performance.

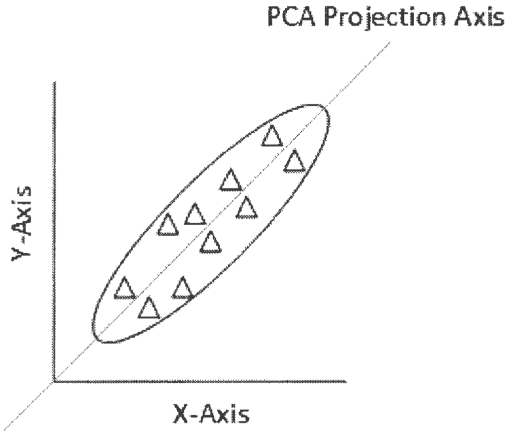


Figure 2.7: Input data are projected to the PCA projection axis

The PCA projection matrix is obtained by solving the eigenvector problem of the covariance matrix of the input matrix. Let A be the input matrix where $A = \{p_1, p_2, \dots p_n\}$. In addition, the mean of each input vector, p_i, is 0 (A is zero mean). C is the covariance matrix of A defined as

$$C = AA^T \tag{2.4}$$

The eigenvector and eigenvalue of C are defined as

$$Cx = \lambda x \tag{2.5}$$

where x represents one eigenvector of C and λ is its corresponding eigenvalue. The vector, z_i, is one principle component of A, which is computed by

$$z_i = x_i^T A \tag{2.6}$$

The variance of z_i is the corresponding eigenvalues λ_i. PCA keeps the first n principle components with the largest variance while throws away others which are regarded as noise. The transformation matrix W is formed by the first n corresponding eigenvectors which is presented as

$$W_{PCA} = [x_1, x_2, \ldots x_n] \tag{2.7}$$

The projection of the input matrix A in PCA subspace is then computed as

$$A_{PCA} = W_{PCA}^T A \tag{2.8}$$

It can be proved that PCA minimizes the reconstruction error, $||A - W^T A||$.

For each input vector p_i, the projected new vector is $p_i\prime = W_{PCA}^T p_i$. For input with non zero mean value, data must be centred to 0 by subtracting mean. Equation 2.3 is thus rewritten as

$$p\prime = W_{PCA}^T (p - \overline{p}) \tag{2.9}$$

where $\overline{p}$ denotes the mean of the vector p. The dimensionality of $p\prime$ is generally lower than p, where $p\prime \in \mathbb{R}^{n \times 1}$ (n is the number of principle components retained in W_{PCA}).

2.2.2 PCA Whitening

According to previous discussion in Section 2.2.1, the projection result of PCA is composed by dimensions with highest variance. It is reasonable in some applications but in face recognition the high variability of image usually corresponds to illumination, facial expression, etc. It has been suggested that removing the most significant three dimensions to form the transformation matrix can reduce the variation due to lighting [8]. Recent works [28, 24, 29, 30, 31] suggest to normalize all the components by whitening. This approach assumes the discriminative information is distributed equally among all dimensions, thus the noise can be reduced by downweighting the high variance components whereas increasing the week ones.

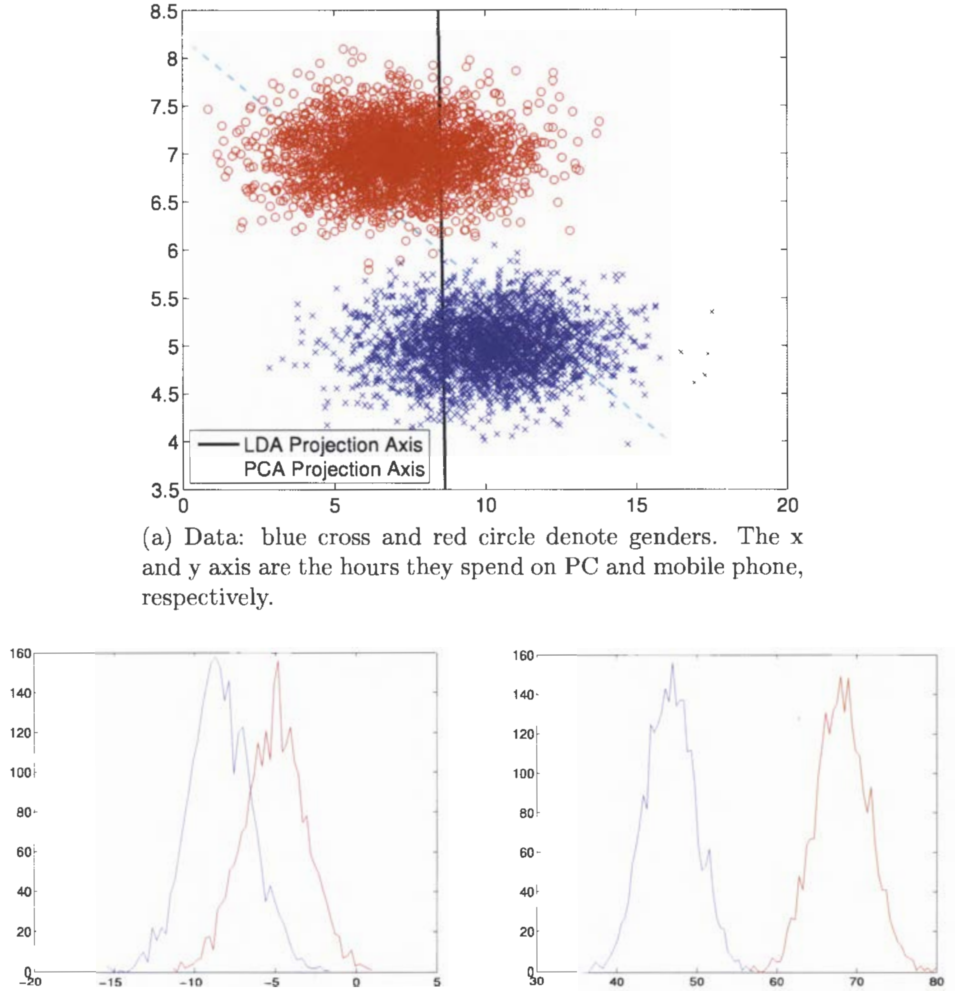
The whitening transformation is a decorrelation transformation in which the output vectors are uncorrelated and have variance of 1. In the case of PCA whitening, the PCA transformation matrix yields to another whitened matrix by

$$W_{WPCA} = \Lambda^{-\frac{1}{2}} W_{PCA}^{T} \tag{2.10}$$

where $\Lambda^{-\frac{1}{2}} = diag(\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}} \ldots \lambda_n^{-\frac{1}{2}})$, λ_i is the corresponding eigenvalue of the eigenvector in W_{PCA}.

2.2.3 Linear Discriminant Analysis

Unlike PCA which is a unsupervised learning technique, LDA is a supervised learning method. It searches to project inputs into a subspace that preserve maximum discriminatory information. The similarities and differences of LDA and PCA are briefly



(a) Data: blue cross and red circle denote genders. The x and y axis are the hours they spend on PC and mobile phone, respectively.

(b) The distribution after PCA projection. x axis represents projected data and y axis represents the number of data projected into this area.

(c) The distribution after LDA projection. x axis represents projected data and y axis represents the number of data projected into this area.

Figure 2.8: The difference of PCA and LDA projection. It can be seen that PCA keeps more variance of the data (curves are wider than in LDA) whereas LDA separates the samples with a larger margin. The data with different label are hard to differentiate in PCA; however, solving the LDA problem is not always feasible due to lack of training data.

demonstrated in Figure 2.8. If we design a system to predict the gender of people according to his/her behaviour, the LDA can provide better discriminative information than PCA.

Formally, LDA finds an optimal projection matrix W_{LDA} which maximizes the equation below

$$W_{LDA} = \arg\max \frac{W^T S_B W}{W^T S_W W} \tag{2.11}$$

where S_B is the between classes scatter matrix and S_W is the within classes scatter matrix. The between-class scatter matrix is defined as

$$S_B = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2.12}$$

And the within-class scatter matrix is defined as

$$S_W = \sum_{c} \sum_{x \in c} (x_i - \mu_i)(x_i - \mu_i)^T \tag{2.13}$$

where μ_i is the mean of i-th class and n_i is the samples number of this class, μ the overall mean of all classes, c is the total number of classes. The Equation 2.11 is solved by the generalized eigenvalue problem expressed as

$$S_B W = \lambda^T S_W W \tag{2.14}$$

Then W is formed by the first n eigenvectors of matrix $S_W^{-1} S_B$. However, in face recognition system the S_W is often singular (i.e., S_W^{-1} does not exist) due to the high dimensionality of facial features. [8] suggested to use PCA before LDA to reduce its dimensionality to avoid such singular problem.

2.3 Classifier

Three commonly used classifiers, namely, Nearest Neighbour (NN) classifiers, Support Vector Machines (SVM) and Convolutional Neural Network (CNN) are introduced in this section.

2.3.1 Nearest Neighbour Classifiers

There are many possible distance measures available for classification purpose. Some of them are presented below.

Euclidean distance:

$$d(p,q) = \sqrt{\sum (p_i - q_i)^2} \tag{2.15}$$

where $p = [p_1, p_2, \dots p_n], q = [q_1, q_2, \dots q_n]$.

Histogram intersection:

$$d(p,q) = \sum \min(p_i, q_i) \tag{2.16}$$

Log-likelihood statistic:

$$d(p,q) = -\sum p_i \log q_i \tag{2.17}$$

Chi square statistic (χ^2):

$$d(p,q) = \sum \frac{(p_i - q_i)^2}{p_i + q_i} \tag{2.18}$$

Cosine similarity:

$$d(p, q) = \frac{p^T q}{||p|| \, ||q||} \tag{2.19}$$

Mahalanobis distance:

$$d(p, q) = \sqrt{(p - q)^T S^{-1} (p - q)} \tag{2.20}$$

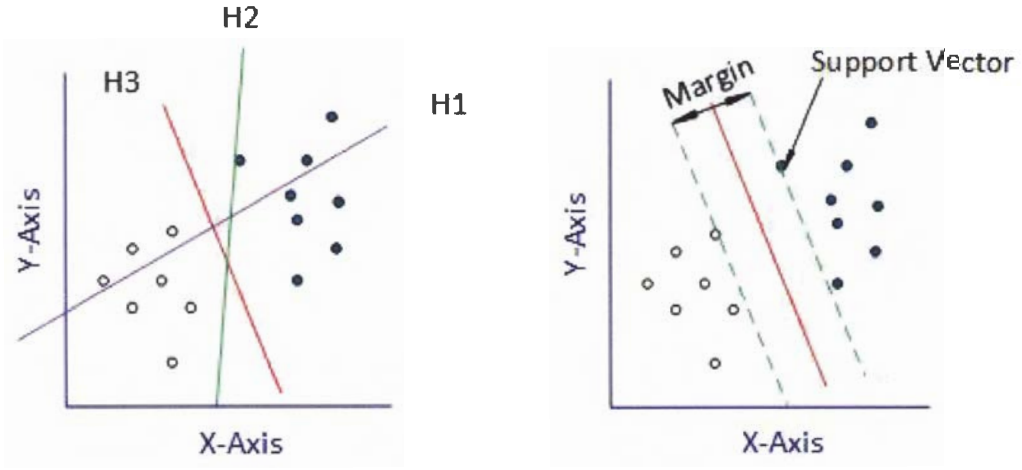
where S is the covariance matrix. Although few methods use the regular Mahalanobis distance as classifier, a category of classifiers named matrices learning extend this equation by employ learn-based Mahalanobis matrix S to compute the distance.

2.3.2 Support Vector Machines

Support Vector Machines (SVM) is a supervised learning algorithm used for binary classification. Given a set of training samples with labels $\{x_i, y_i\}$ where x_i denotes the sample vector and $y_i \in \{-1, 1\}$ is class label, SVM seeks a hyper plane $\mathbf{w} \cdot \mathbf{x} - b = 0$ in hyperspace $\mathcal{H}$ which separates the samples according to their labels ((a) in Figure 2.9). Here $\mathbf{w}$ is the weight vector and b is the bias.

By varying the bias, we can obtain infinite number of hyper planes with the same weight vector. Among all the possible representations of the hyper plane, the maximal and minimal value of b present such kind of hyper planes that it minimizes the distance between the hyper plane and samples from one class. Here the sample(s) which is closest to the hyper plane is called support vector.

As a matter of convention, we scale $\mathbf{w}$ and b to proper values to represent these



(a) H_1,H_2 and H_3 represent three hyper plane to separate classes. H_1 fails on the separation, while H_2 separates them successfully. H_3 is the optimized separation among all.

(b) Among all the possible hyper planes, two of them minimize the distance between the hyper planes and observations from one class. The distance between these two hyper planes is called margin. The samples in the hyper planes are called support vectors.

Figure 2.9: Linear separation of inputs

two hyper planes as:

$$\mathbf{w} \cdot \mathbf{x} - b_1 = 1 \tag{2.21}$$

$$\mathbf{w} \cdot \mathbf{x} - b_2 = -1 \tag{2.22}$$

The distance between these two hyper planes is $\frac{2}{\|w\|}$ which is called margin. Intuitively, the optimized separation should maximize the margin when we have no prior knowledge of the distribution ((b) in Figure 2.9).

Then the optimization problem can be written as

$$\min L(w) = \|w\| \,\text{subject to}\; y_i(w \cdot x_i - b) \geq 1 \tag{2.23}$$

This is a problem of Lagrangian optimization and can be solved using Lagrange multipliers α_i.

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b) \tag{2.24}$$

where α_i and b are found by using SVC learning algorithm [32] and

$$sgn(v) = \begin{cases} 1, \text{if } v \geq 0 \\ -1, \text{if } v < 0 \end{cases} \tag{2.25}$$

For the linear separation in the input space, $K(x_i, x) = x_i \cdot x$. For non-linear separation, the technique called kernel trick is employed. Samples are projected into feature space which is linearly-separable (same as Figure 2.1). Some of popular kernels are presented as follows.

Polynomial kernel:

$$K(p, q) = (p^T q + c)^d \tag{2.26}$$

Radial basis function (RBF) kernel:

$$K(p, q) = \exp(-\frac{\|p - q\|}{2\sigma^2}) \tag{2.27}$$

Sigmoid kernel:

$$K(p, q) = \tan(\gamma p^T q + c) \tag{2.28}$$

2.3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a non-linear classifier which is inspired by biological neural network. Unlike other classifiers above, CNN usually perform classification on intensity images. It extracts and classifies features in the same framework. The details of CNN will be given in Section 3.3.1.

Chapter 3

Previous Works

In this chapter, previous works which this framework is based on or is inspired by will be introduced. In Section 3.1 and 3.1.1, algorithms based on over-complete feature and patches are introduced, which are used in the feature extraction stage our proposed method. Next, in the third section, a regular Convolutional Neural Network architecture is described, as well as the borrowed ideas from this architecture that we are use to build the newly proposed method.

3.1 Over-Complete Feature

Instead of designing a new computer vision feature from scratch, some recent efforts [23, 25, 33, 34, 35, 36, 22], have particularly focused on extracting over-complete feature with off-the-shelf LBP descriptors.

Over-complete features are extracted from the image in a redundant and heavily overlapped way. Comparing with regular feature, it is more informative. The algorithms discover the invariant pattern across all feature to extract more robust discriminative features. However, the features are also of high-dimensionality because they contain redundant information. Therefore, feature compression techniques are desirable in this kind of algorithm. In the rest of this section, several over complete feature extraction and compression methods are introduced.

3.1.1 Feature Extraction

Several schemes are used to reform a regular feature extraction algorithm to extract over-complete feature, which is summarized in the follows.

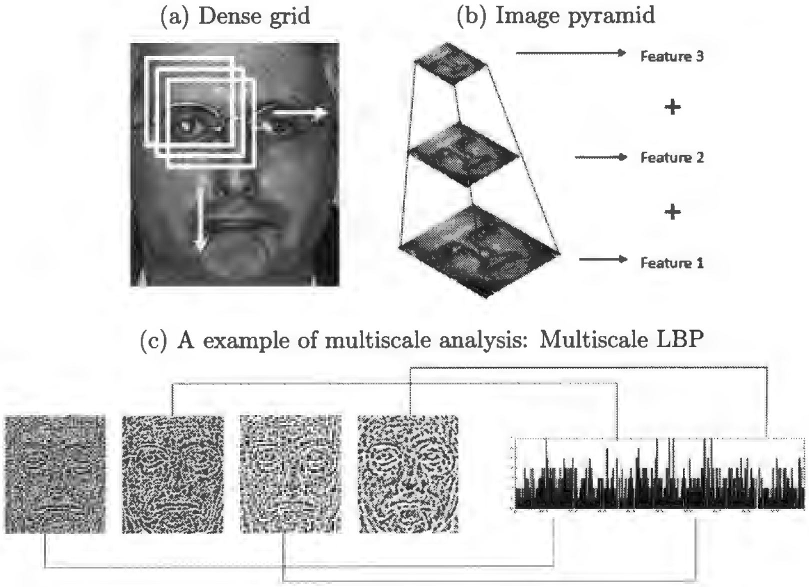
Dense grid: For computer vision descriptors which are computed from grids, their dense versions can be obtained by forcing such grids to heavily overlap. Densely extracted SIFT [33, 34, 35, 36] and LBP [33, 25] fall into this category.

Image pyramid: By using image pyramid, the original image is scaled to different size to extract features in different resolutions [37, 23]. The features from higher level capture globe structure information whereas the later level is able to extract detail texture of the image.

Multiscale analysis: The third way is by adopting multiscale analysis framework. The dense feature is obtained varying one or multiple parameters of the original algorithm and fusing them into one high level feature. For instance, Gabor features are obtained by applying a family of Gabor wavelets [21, 38]; [22] proposed a multi-scale LBP descriptor which combines features computed by multiple LBP operators.

Some systems may employ more than one schemes to obtain the over-complete features. The three discussed schemes are presented in Figure 3.1.

Figure 3.1: Schemes of over-complete feature extraction



(a) Dense grid

(b) Image pyramid

(c) A example of multiscale analysis: Multiscale LBP

3.1.2 Feature Compression

The dense feature contains a lot of redundant information. Thus it can be compacted into a smaller size without losing much information. In addition, the dimensionality reduction can also benefit the recognition rate by removing unimportant information which is usually noise and expose high-level transformation invariant features. Some remarkable approaches are summarized as follows.

Subspace projection

The details of subspace projection have been discussed in Section 2.2. To compress feature by subspace projection, the first step is to stack all the dense features into one vector. Next, the transform matrix is learned to reduce the dimensionality of the stacked features. In Section 3.1.1, [23] uses PCA and [22] uses PCA+LDA to reduce the dimension of their features respectively.

Fisher Vector

Fisher Vector (FV) encodes large set of features into one high dimensional vector by Gaussian Mixture Models(GMM) [39, 36]. GMM is a soft assignment algorithm which aims to find K Gaussian components that minimize the overall probability of each samples presented in the Equation 3.1

$$P(x) = \sum_{i=1}^{K} w_k \mathcal{N}(x|\mu_i, \sigma_i) \tag{3.1}$$

where x is the input features, w_k is the weight of the Gaussian component, $\mathcal{N}(x|\mu_i, \sigma_i)$ is the multivariate Gaussian component with mean μ_i and covariance σ_i. The GMM problem is solved by expectation-maximization (EM) algorithm [40, 41].

After solving the GMM problem, the mean, $\phi_k^{(1)}$, and covariance deviation, $\phi_k^{(2)}$, (the average first and second feature differences) between features and each of the

GMM centres are computed by:

$$\phi_k^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_p(k)(\frac{x_p - \mu_k}{\sigma_k}) \tag{3.2}$$

$$\phi_k^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_p(k)(\frac{(x_p - \mu_k)^2}{\sigma_k^2} - 1) \tag{3.3}$$

where N is the total numbers of features and $\alpha_p(k)$ is the soft assignment weight of p-th feature x_p to the k-th Gaussian. Finally, the Fisher Vector feature of one image is obtained by stacking all the results into one vector:

$$\phi = [\phi_1^{(1)}, \phi_1^{(2)}, \ldots \phi_K^{(1)}, \phi_K^{(2)}] \tag{3.4}$$

This feature represents the differences between this particular image and the distribution of all the training images in feature space. Note that although this representation is still of high dimensionality (65536 for $K = 512$, feature length $= 64$), it is significantly lower than directly concatenation all the obtained features (1.7M in the case of [36]).

Probabilistic Elastic Matching

Li *et al.* proposed a method named Probabilistic Elastic Matching which has a similar approach to Fisher Vector and achieved a even better result [33, 34, 35]. Similar as FV, a GMM is trained from the dense features. They force the covariance matrix of each GMM to be spherical to balance the two parts of extracted features (the appearance feature and its spatial location, respectively). After fitting into GMM, the features are encoded by the highest probability of all the features from one image to each GMM centres, formally,

$$n_k = \arg\max \mathcal{N}(x_p|\mu_k, \sigma_k) \tag{3.5}$$

And the compact representation of this image is

$$n = [n_1, n_2, \ldots n_k] \tag{3.6}$$

It should be noted that the dimensionality of this feature is lower than FV since n_i is scalar while ϕ_i is matching difference vector.

3.2 Patch-based Algorithm

The holistic face recognition algorithm take the whole face image as input and generate only one facial descriptor, such as Eigenface and Fisherface [8]. On the contrary, the patch-based system measures the similarity using a divide-and-conquer strategy. The image is firstly partitioned into several patches to extract regional features. The final decision is made by considering the similarities among all patches. The patch-based system has inherent advantage on variations of lighting, facial expression or occlusion, since these variations are always smaller in the patch than in the whole face.

3.2.1 Naive Patch-based Algorithm

In the naive patch-based algorithm, the image, u, is firstly partitioned into several non-overlapped patches whose coordinate is denoted as $\mathcal{P}_{i,j}(u)$. To compare two images

u and v, the similarity, $s_{i,j}$, for each corresponding pairs of patches of two images is calculated as:

$$s_{i,j} = d(\mathcal{P}_{i,j}(u), \mathcal{P}_{i,j}(v)) \tag{3.7}$$

where d is the dissimilarity measure. The overall distance is then computed by combining all regional distance. The most straightforward way is to sum all of them as follows

$$S = \sum w_{i,j} s_{i,j} \tag{3.8}$$

where $w_{i,j}$ is the weight of $\mathcal{P}_{i,j}$ patch. If no realizable ways exist to determine this parameter, it can be set empirically or leave it as 1.

This naive algorithm is simple and fast, thus it is adopted by many works [38, 19, 22]. However, its performance is not optimal. Many advanced patch-based frameworks have been proposed, and two of them are introduced in the rest of this section.

3.2.2 Electoral College

Chen *et al.* proposed a unified framework named electoral college [42] which mainly tackle the misalignment problem. In this method, every patches in the images, u, in gallery set allow to shift in a small range s to form a pile of patches which is defined by

$$\mathcal{R}_{i,j}(u) = \{\mathcal{P}_{i-s,j-s}(u), \mathcal{P}_{i-s+1,j-s}(u), \ldots \mathcal{P}_{i+s,j+s}(u)\} \tag{3.9}$$

The regional similarity is defined as the highest similarity between the pile of patches from image u and patch from image v, i.e.

$$s_{i,j} = \max(d(\mathcal{R}_{i,j}(u), \mathcal{P}_{i,j}(v))) \tag{3.10}$$

This framework is proved to be able to improve all the holistic algorithm [42] together with the original LBP approach [43].

3.2.3 Component-level face alignment

A different way of generating patches is suggested in [29]. In their work, the patches are generated to represent a specific component of the face. They identified 9 such kind of components in total, including forehead, left eyebrow, right eyebrow, left eye, right eye, nose, left cheek, right cheek and mouth. The regions of components are obtained by the facial landmarks (e.g., eyes, nose, mouth), for example, the forehead component is obtained by cropping a particular region of image aligned by left eye and right eye. Note that unlike naive algorithm and electoral college, patches in this framework are not in the same size and are partially overlapped. They argued that the large pose variation can be handled since the facial component can be aligned more accurately.

3.3 Deep Learning

Deep learning is a methodology of machine learning for obtaining hierarchical representations of images in our case. It uses a multilayer cascade to extract high level abstraction or representations of data. The features extracted in earlier layer are fed to the later layer as input. Additionally, feature extraction is in a heavily overlapped manner. Feature in earlier layer can contribute to multiple features in the later layer. In the rest of this section the most well-studied deep learning architecture, Convolutional Neural Network, is introduced.

3.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is variant of neural network pioneered by [44], improved by [5] and simplified by [45]. It is a biologically-inspired architecture which contains multiple interconnecting neurons to form a network. CNN is proved to have inherent advantage in computer vision processing because of its 2-dimensional convolutional kernels and feature maps. As shown in Figure 3.2, layers in CNN hierarchically extract higher level features from their input layers, and then connect to multilayer perceptron neural network (MLP) to perform classification. The extracted features in earlier layer focus on detail information such as edge and corner, while the features in the latter layer represent a higher level of abstraction. Several different kinds of layers common in all CNN architectures are described as follows.

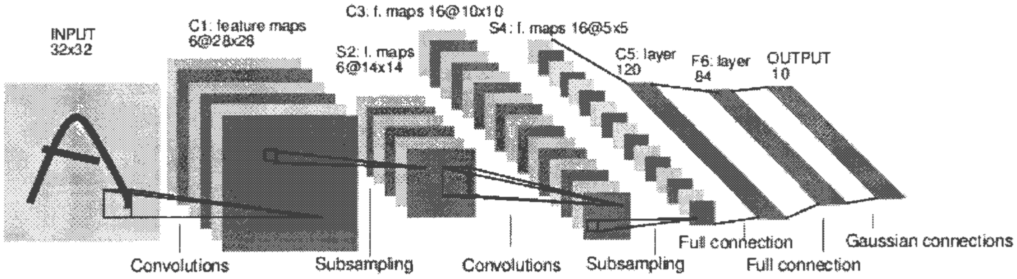


Figure 3.2: One example of Convolutional Neural Network: the architecture of LeNet-5 from [5].

Convolutional Layer: The convolutional kernel (filter) works as neuron in this layer. The neuron is defined as

$$a^{l+1} = \alpha(b + w * a^l) \tag{3.11}$$

where a^l is the input and w is the corresponding weight, $*$ is denoted as the convolutional operator, b is the bias and a^{l+1} is the output. Here α is called activation function which provides nonlinearity.

To simplify the learning process, kernels with the same parameters replicated over the entire image/feature map and connect to the latter layer with same weight. In addition, multiple kernels exist in one layer to generate different feature maps. Another important characteristic of this layer is that convolutional kernels that only connect particular subset of feature maps from earlier layer. This helps to further reduce the computational cost.

Subsampling (Pooling) Layer: The purpose of subsampling layer is to reduce the size of feature map to consequently reduce variance. The feature map is spited into several non-overlapped cells, and the maximum or mean feature of this cell is taken as output to form pooled features.

Fully Connected Layers: The fully connected layers are a MLP connecting to the convolutional part of CNN to perform the classification. Unlike convolutional layers, these layers fully connect to their earlier layers. The size of the feature map keeps reducing with every new convolution and sampling operation in hierarchical layers. With proper selected parameters, the size will reduce to 1×1, thus all the feature maps can be stacked into one single vector as the input to MLP.

CNN employs the backpropagation algorithm with gradient descent for learning purpose. To begin with, CNN computes the output of each input samples and finds out the output error by comparing it with the correct label. After that, the error is backpropagated to each layer to compute the gradient of the error. Finally, the parameters of CNN are adjusted according to the gradient. This approach repeats until convergence is reached.

Although many techniques such as shared weight are employed to simplify the model, CNN is still an architecture of high complexity and consequently is hard to train. First, the learning approach itself is computational expensive. With the increasing number of processing layers, the gradient tends to be unstable which means the parameters are hard to converge. This issue is partially solved by using more powerful computer hardware, especially by leveraging GPU computing techniques. Second, since there exist thousands or sometimes millions of parameters in one CNN, the model is easily overfitted. Therefore, a large amount of training data is desired. However, regarding to the face recognition, the available samples are usually quite limited. One common solution is to learn the the networks with outside data. For example, all the CNN which achieve state-of-the-art in the LFW benchmark are learned by massive outside data [9, 10, 11, 12, 13, 14].

Chapter 4

Proposed Algorithm

In this chapter, a simple but powerful deep learning architecture called Local Binary Pattern Network (LBPNet) is proposed so as to provide a novel tool for face recognition. In the rest of this chapter, we will firstly describe the LBPNet architecture in details. Next, an introduction on video-based face recognition on LBPNet will be given.

4.1 Architecture

The architecture of LBPNet can be divided into two parts: (i) deep network for feature extraction, and (ii) regular network for classification. The overall diagram of our proposed system is shown in Figure 4.1.

Two layers in the deep network, which use LBP and PCA filter respectively, are

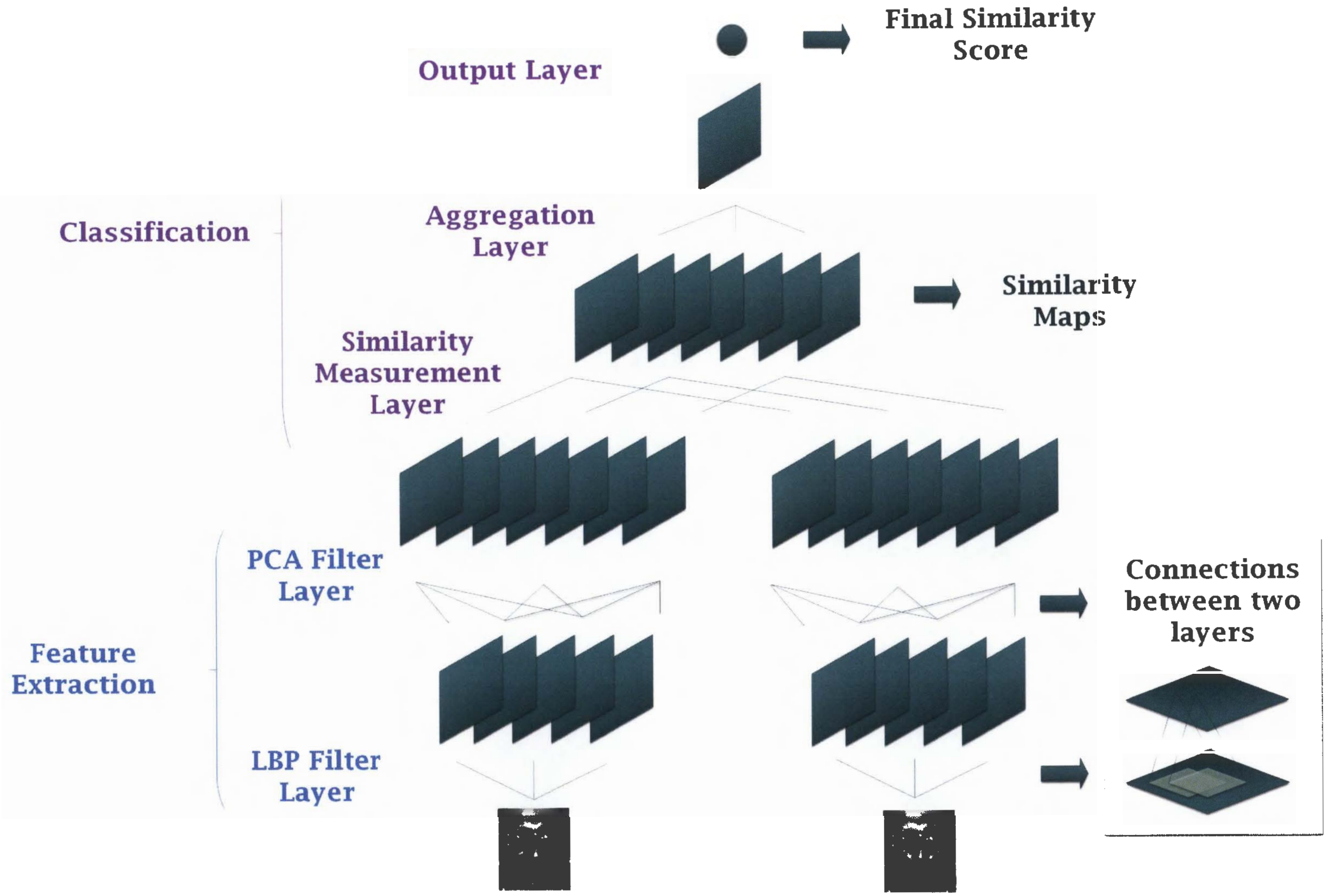


Figure 4.1: A schematic diagram of LBPNet.

hierarchically connected to extract high-level over-complete representations of the images. Two such networks are connected with the classification networks, allowing taking two images as the input. Hence, the similarity measurement can be performed based on the extracted features. The decision is made according to the output of the network which is the similarity of these two images: in identification system, the face with the highest similarity in gallery set is chosen as the identity of the probe image; in verification system, the hypothesis is accepted or rejected by thresholding the similarity value. Details of each layer are described in the following subsections.

4.1.1 LBP Filter Layer

A filter in image processing is a neighbourhood operation, of which the output is computed by applying an algorithm to the values of the pixels in the neighbourhood of the corresponding input pixel. In the LBP filter layer, the filters are based on LBP operator described in [19]. The LBP operator, $LBP^{u2}_{P,R}$, labels each pixel g_c in the image by thresholding its P surrounding points g_p ($p \in [1, P]$) and stacking labels l_p (defined in Eq. 4.1) into one binary string

$$l_p = \begin{cases} 1, \text{when } g_p > g_c \\ 0, \text{when } g_p \leq g_c \end{cases} \tag{4.1}$$

The points are sampled from a circle of radius of R, whose centre is at g_c. In addition, we use the unique pattern (denoted as $u2$ in the operator) to encode the labels. The feature generated in the filter is formulated as

$$h_{P,R,z}(i) = (\sum B((LBP^{u2}_{P,r}(x, y) = i)))^{-\frac{1}{2}} \tag{4.2}$$

where z is the size of the filter, and $B(v)$ is 1 when v is true; 0 otherwise.

Note that here the square root of the LBP histograms is used to increase the discrimination ability [28]. By replicating the kernel, a 3-dimensional feature cube is generated from the image.

To capture multi-scale representations of the image, the computation is repeated in the LBP filter layer using multiple kernels subject to different combinations of LBP radius, r and filter size, z. The features obtained in this layer represent the multi-scale LBP histogram features of the image. When considering them as one feature vector, it is of high dimensionality, which can be over $10M$ in our experiments.

4.1.2 PCA Filter Layer

The objective of this layer is to generate outputs from the input features which is both lower in dimensionality and higher in the capability of abstraction. In the context of face recognition, the outputs represent multi-scale patch-based features. The objective is mainly achieved by performing PCA on each computation window.

To start with, the input features are sampled and concatenated into the vector p_z, which is given by

$$\begin{aligned} p_z = &[h_{r_1,z}(u,v), h_{r_1,z}(u+s,v), \ldots \\ &h_{r_k,z}(u+n\times s, v+n\times s)] \end{aligned} \tag{4.3}$$

$$n = \lfloor M/s \rfloor \tag{4.4}$$

where the size of the filter is $M \times M$, s is the sampling stride, $h_{r,z}(u,v)$ is the feature vector located at the u-th column and the v-th row of the feature cube generated by the LBP filter with sample radius r and size z. (u, v) denotes the starting point of sampling. Considering the feature extraction is in a dense grid in the earlier layer, the features are highly redundant —in general, two neighbourhood features can share up to 90% of the same LBP labels. Therefore, features are sampled to reduce the resulting vector length while preserving the critical discriminative information.

Here, the PCA filter only computes based on the feature cubes that are generated by LBP filters of the same size. This resembles the partial connections between convolutional layers in CNN. We find it helps simplify the computation and increase the discrimination ability.

After obtaining the concatenated vector, we reduce the dimensionality by PCA projection. In general, for a given matrix A, PCA seeks a transformation matrix, W, which minimizes the reconstruction error, $||A - W^T A||$. The solution is known as the matrix constructed by the first n eigenvectors of the covariance matrix $C = A^T A$ when A is zero mean. In our case, the input matrix, A, is formed by

$$A = [q_1, q_2, \ldots, q_n]^T \tag{4.5}$$

where q_i is defined as

$$q_i = p_i - \frac{1}{J} \sum_{j=1}^{J} p_j \tag{4.6}$$

In this method the extracted feature (p_i) yields another vector (q_i) by subtracting the mean vector of the entire training set from itself.

In the context of face recognition, high variability of images generally corresponds to the change of illumination, facial expression, etc. Such impact of high variability can be reduced by downweighting the high variance directions whereas increasing the weak ones, since the discriminative information are uniformly distributed over all directions of the data [24]. Here, we normalize all the eigenvectors by whitening, the transformation matrix is then expressed as

$$W = [\lambda_1^{-\frac{1}{2}} x_1, \lambda_2^{-\frac{1}{2}} x_2, \ldots \lambda_n^{-\frac{1}{2}} x_n] \tag{4.7}$$

where λ_i is the eigenvalue of the corresponding eigenvector x_i. The output feature is then extracted by

$$\sigma_i = W_i^T q_i \tag{4.8}$$

where W_i is the whitened PCA transformation matrix. Same as the first layer, multi-scale representation can be obtained using multiple filters with different parameters. Here we vary the starting point, (i, j), and leave other parameters unchanged. Figure 4.2 shows an example of 4 filters with different starting points. The diagram of the deep part of LBPNet consisting of the first two layers is presented in Figure 4.3. The outputs of the deep network represent patch-based over-complete features of the image.

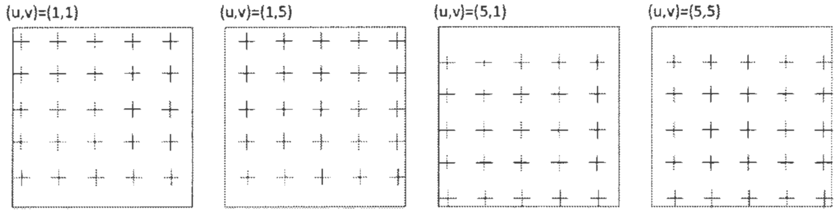


Figure 4.2: 4 different PCA filters. Only vary the starting point of sampling and keep other parameters unchanged.

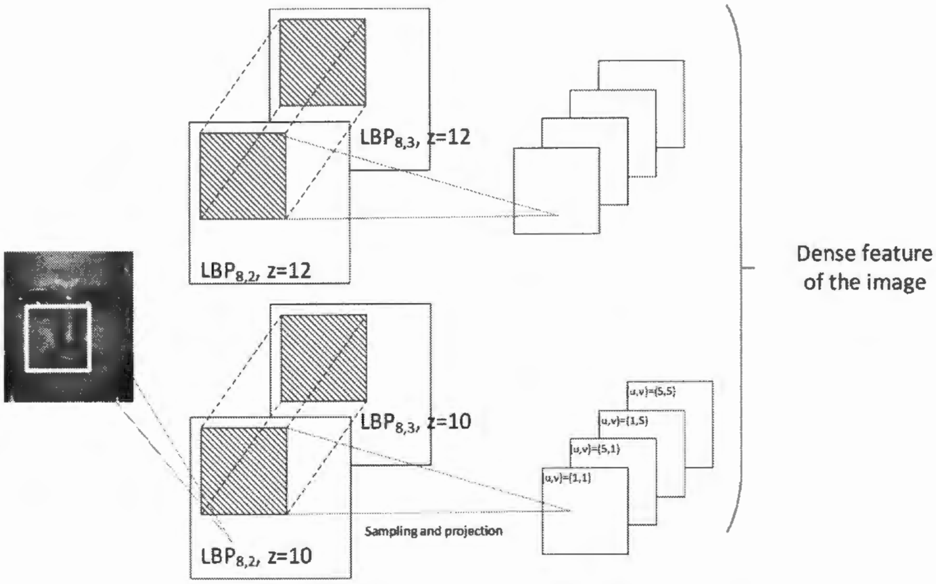


Figure 4.3: The deep part of LBPNet. The feature cubes are represented as 2-dimensional map.

4.1.3 Similarity Measurement Layer

Two deep networks are connected to accept two images as the input. The extracted features in the upper layer consist of two subsets from each image, respectively. The regional similarity scores, δ_i, are computed pairwisely between two corresponding features. Here we use angle-based measure, cosine similarity, which is formulated as

$$\delta_i = \frac{\sigma_i \cdot \sigma\prime_i}{\|\sigma_i\| \, \|\sigma\prime_i\|} \tag{4.9}$$

where $\sigma_i, \sigma\prime_i$ are two features from upper layers, respectively. The output of these layers represents the regional similarity of two faces in specific scale.

4.1.4 Aggregation Layer

In this layer, we reduce the number of regional similarities before training the network. It is assumed that the similarities of the same coordinate in different map contribute equally to the final score, then all the maps are aggregated into one map by

$$\lambda_i = \frac{1}{J}\sum_{j=1}^{J}\delta_{i,j} \tag{4.10}$$

where $\delta_{i,j}$ is the i-th score in j-th map, and J represents the total number of maps.

4.1.5 Output Layer

The unsupervised learning is the deep learning part of LBPNet. To provide unsupervised learning on the output layer, the output λ is computed as

$$\Lambda = \frac{1}{I}\sum_{i=1}^{I}\lambda_i \tag{4.11}$$

where I represents number of patches. This method uses the average of all regional similarities as the overall similarity. However, the performance can further boost by training this layer in supervised manner. It is done by assigning different weight a_i for each λ_i to compute the overall similarity:

$$\Lambda = \frac{1}{I}\sum_{i=1}^{I}a_i\lambda_i \tag{4.12}$$

The coefficients, $\{a_1, a_2, \ldots a_I\}$, should maximize the similarity, Λ, between same people and minimize it between different people. In practice, we use linear-SVM to determine the coefficients.

4.2 LBPNet for Video Based Recognition

Although LBPNet is initially designed for still image face recognition, it is also possible to use it for video based tasks. The naive algorithm is to use all frames in video as gallery set. However, the number of available frames is too high that this algorithm is computational unfeasible. Following [46], after aligning all faces to the same position, we average the LBP features of them in the first layer to form a mean feature vector:

$$h_k = \frac{1}{L}\sum_{l=1}^{L} h_k^*(l) \tag{4.13}$$

where $h_k^*(l)$ is the k-th LBP feature in the l-th frames and h_k is the mean feature of k-th cell in the video clip. Once the mean feature vector is generated, it can be used as the feature from still image in LBPNet.

Chapter 5

Experiment Design

We experimentally validate our framework on the public benchmarks FERET [47], LFW [1, 2] and YTF [4] datasets. In this chapter, the database as well as the experiment setting will be introduced.

5.1 Experiment on Face Identification: FERET

To evaluate the capability of LBPNet on face identification, we use one of the well known Face Recognition Technology (FERET) [47] dataset. This dataset contains controlled images of $1,196$ individuals. It contains one gallery set and 4 probe sets: (i) Fb set, which is taken in the same condition but with different facial expression, (ii) Fc set, which is taken in different light condition, (iii) Dup-I set, which is taken between one minute and 1031 days after the gallary set, (iv) Dup-II set, which is a subset of Dup-I and is taken after at least 18 months.

Figure 5.1: Examples from FERET dataset. Pictures in the upper line are probe images and the pictures beneath them are the matching ones on gallery set.

The original FERET dataset is provided with a ground truth information file where eyes positions are recorded. We use CSU tools to perform the face normalization and crop the centre region of 150 × 130 according to this file. The images are also preprocessed following the suggestion by Tan *et al.* [48]. All the parameters in this experiment are listed in the Table 5.1.

Table 5.1: Parameter settings for experiment on FERET

LBP filter	
LBP operators	$\{LBP^{u2}_{2,8}, LBP^{u2}_{3,8}, LBP^{u2}_{4,8}\}$
LBP filter size	$c = \{11, 12\}$
PCA filter	
PCA filter size	$w = 110$
sampling stride in the window	$s_1 = c$
starting point of sampling	$i, j \in \{1, c/2\}$
PCA dimension	$d = 1800$
stride of the PCA filter	$s_2 = 10$

5.2 Experiment on Face Verification: LFW

For the face verification task, the *de-facto* evaluation benchmark, Labeled Faces in the Wild [1, 2] dataset, is used to evaluate our framework. LFW is an image dataset for unconstrained face verification which contains 13,233 images of faces of 5,749 individuals. Each face has been labelled with the name of the person pictured. We use the view 2 of dataset which comes with a 10-fold split for cross validation.

We conduct two different experiments on LFW: experiment under unsupervised in which the model is trained without knowing the label information and without the outside data; experiment under restricted setting in which we train our classifier without any outside data.

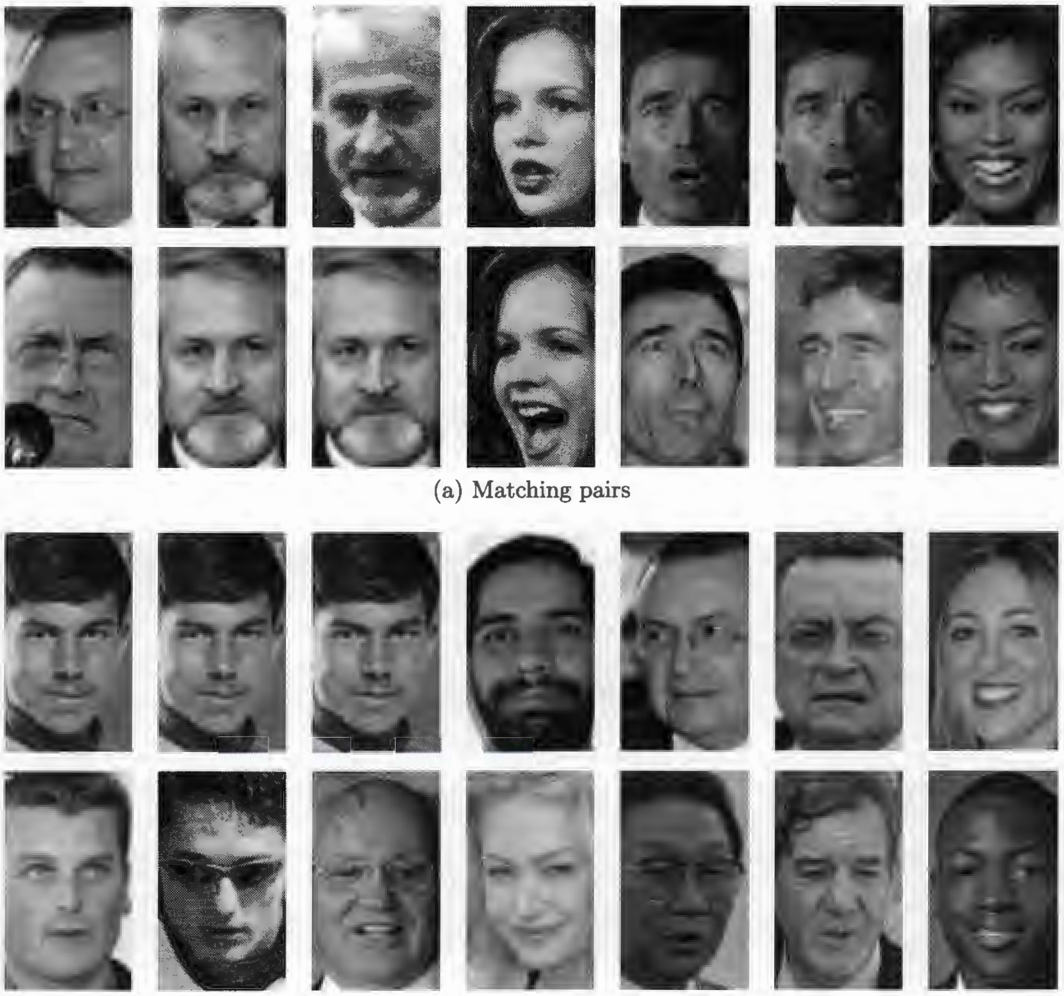
We use the LFW-a dataset which is aligned by commercial software and crop the centre of the images of size 170 × 100. We use the same parameter settings of the LBPNet as the experiment on FERET with some exceptions (Table 5.2))

Table 5.2: Parameter Settings for experiment on LFW

LBP filter	
LBP operators	$\{LBP^{u2}_{1,8}, LBP^{u2}_{2,8}, LBP^{u2}_{3,8}\}$
filter size	$c = \{10, 12, 14, 16, 18, 20\}$
PCA filter	
PCA filter size	$w = 80$
sampling stride in the window	$s_1 = c$
starting point of sampling	$i, j \in \{1, c/2\}$
PCA dimension	$d = 500$
stride of the PCA filter	$s_2 = 10$

5.3 Experiment on Video Based Face Recognition: YTF

To evaluate the capability of our framework on video based face recognition, the popular YouTube Faces (YTF) [4] dataset is used. YTF is a video dataset for un-



(a) Matching pairs

(b) Mismatched pairs

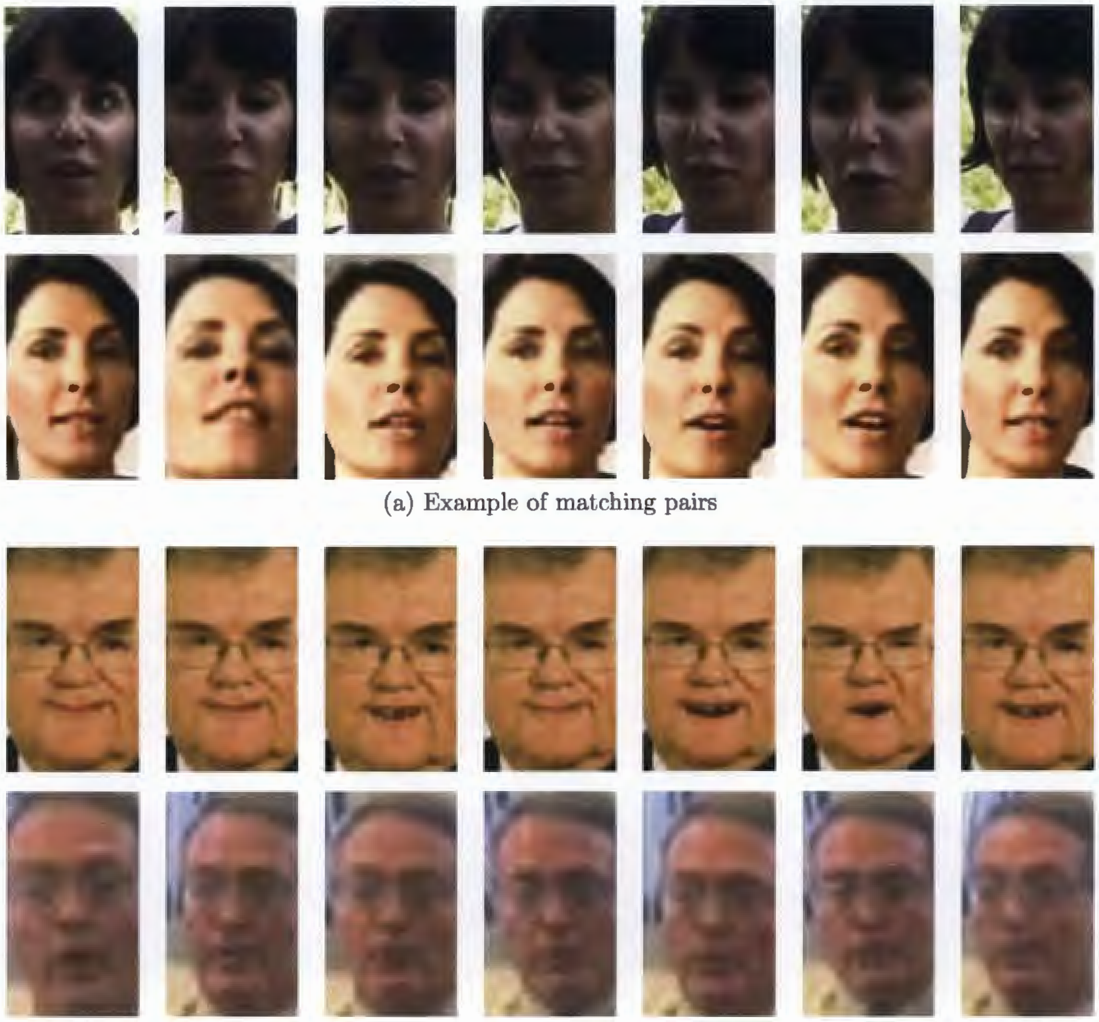
Figure 5.2: These are first 7 matching and mismatched pairs of LFW dataset under view 2. Images are obtained from the provided LFW-a dataset which is aligned version of LFW dataset. The centre regions of size 170 × 100 are cropped.

constrained face verification. The dataset contains 3,425 images of faces of 1,595 individuals whose names come from LFW. Each video contains 181.3 frames on average. Similar as LFW, it comes with a 10 subsets for the restricted protocol. The quality of the picture on YTF is generally worse than LFW.

We use the aligned version of the database and crop the centre region of size 170 × 100 as on LFW. All the parameters are listed in Table 5.2.

Table 5.3: Parameter Settings for experiment on YTF

LBP filter	
LBP operators	$\{LBP^{u2}_{1,8}, LBP^{u2}_{2,8}, LBP^{u2}_{3,8}\}$
filter size	$c = \{12, 14, 16\}$
PCA filter	
PCA filter size	$w = 80$
sampling stride in the window	$s_1 = c$
starting point of sampling	$i, j \in \{1, c/2\}$
PCA dimension	$d = 500$
stride of the PCA filter	$s_2 = 10$



(a) Example of matching pairs

(b) Example of unmatched pairs

Figure 5.3: These are the sampled frames from the first matching and unmatched pairs of YTF dataset. Frames are aligned and the centre regions of size 170 × 100 are cropped. The quality of images is significant worse than LFW dataset.

Chapter 6

Results And Analysis

6.1 Results on FERET

Table 6.1 lists the recognition rate of our framework on FERET dataset as well as some other known approaches. All the results are from their original papers. For the purpose of completeness, we list all the methods known to us. However, to fairly evaluate the LBPNet, only the single type descriptor based unsupervised methods are considered comparable to the LBPNet. As shown in the table, the LBPNet obtains 0.978 in mean recognition accuracy. It outperforms the current best by 0.2%. When looking at each particular probe set, the LBPNet achieves closely matched (on Fb), same good (on Fc) or better (on Dup-I and Dup-II) results. On the most challenged Dup-II probe set, the LBPNet suppresses the current best result (91.0%) in 2.6%. It should be noted that: i) although the supervised learning (use a subset of the dataset to learn their model) and fusion descriptor can increase recognition accuracy, the

Table 6.1: Comparative results of various methods on aligned FERET dataset

	Methods	Fb	Fc	Dup-I	Dup-II	Mean[†]	Comments
Supervised or/and fusion methods	LGBPHS [38]	0.94	0.97	0.68	0.53	0.85	Fusion
	Tan&Triggs [26]	0.98	0.98	0.90	0.85	0.95	Fusion+supervised
	MLBP [22]	0.992	0.995	0.900	0.855	0.961	Supervised
	S[LGBP_Mag-LGXP] [49]	0.99	0.99	0.94	0.93	0.97	Fusion+supervised
	sPOEM+POD [50]	0.997	1.00	0.949	0.940	0.980	Fusion
	GOM [51]	0.999	1.00	0.957	0.931	0.984	Supervised
Single descriptors based unsupervised learning methods	LBP [19]	0.93	0.51	0.61	0.50	0.78	
	LBP Template [43]	0.989	0.928	0.760	0.634	0.905	
	LGBPWP [52]	0.981	0.989	0.838	0.816	0.933	
	LBP-DLMA [53]	0.994	0.993	0.887	0.869	0.957	
	POEM [31]	0.996	0.995	0.888	0.850	0.959	
	G-LQP [24]	0.999	1.00	0.932	0.910	0.976	
	LBPNet	**0.996**	**1.00**	**0.942**	**0.936**	**0.978**	

[†] Computed by Fb+Fc+Dup-I since Dup-II is the subset of Dup-I

LBPNet still outperforms most of them; ii) some competitive methods (i.e., [24, 50, 52]) extract descriptors from the Gabor images (extracted by Gabor filters from the intensity images), which may bring advantages comparing with the regular descriptors.

6.2 Results on LFW

Table 6.2 and Figure 6.1 show the results and ROC on LFW under unsupervised setting comparing with other baselines and state-of-the-art results. Only the single descriptor based methods are listed in the tables for fairly comparison. The LBPNet achieves 0.9404 under this setting which is ranked third best among all. It closely matches the first and second best ones, which are 0.9428 and 0.9405 respectively.

Table 6.2: Comparative results of various methods on LFW dataset view 2 with unsupervised setting

Methods	AUC	Alignment
SD-MATCHES [54]	0.5407	
H-XS-40 [54]	0.7547	
GJD-BC-100 [54]	0.7392	
LARK [55]	0.7830	
MRF-MLBP [56]	0.8994	LFW-a + MRF
Pose Adaptive Filter [57]	0.9405	PAF(3D alignment)
Spartans [58]	0.9428	3D Generic Elastic Model
LBPNet	**0.9404**	LFW-a

Regarding to the experiment under restrict setting, the results are reported in Table 6.3 and ROC curve are plotted in Figure 6.2. The LBPNet achieves promising results, whose average accuracy is 0.8772.

It should be note that most state-of-the-art methods employ sophisticated face register modules (e.g., 3D Generic Elastic Model in [58]) or/and classifiers (e.g., Joint

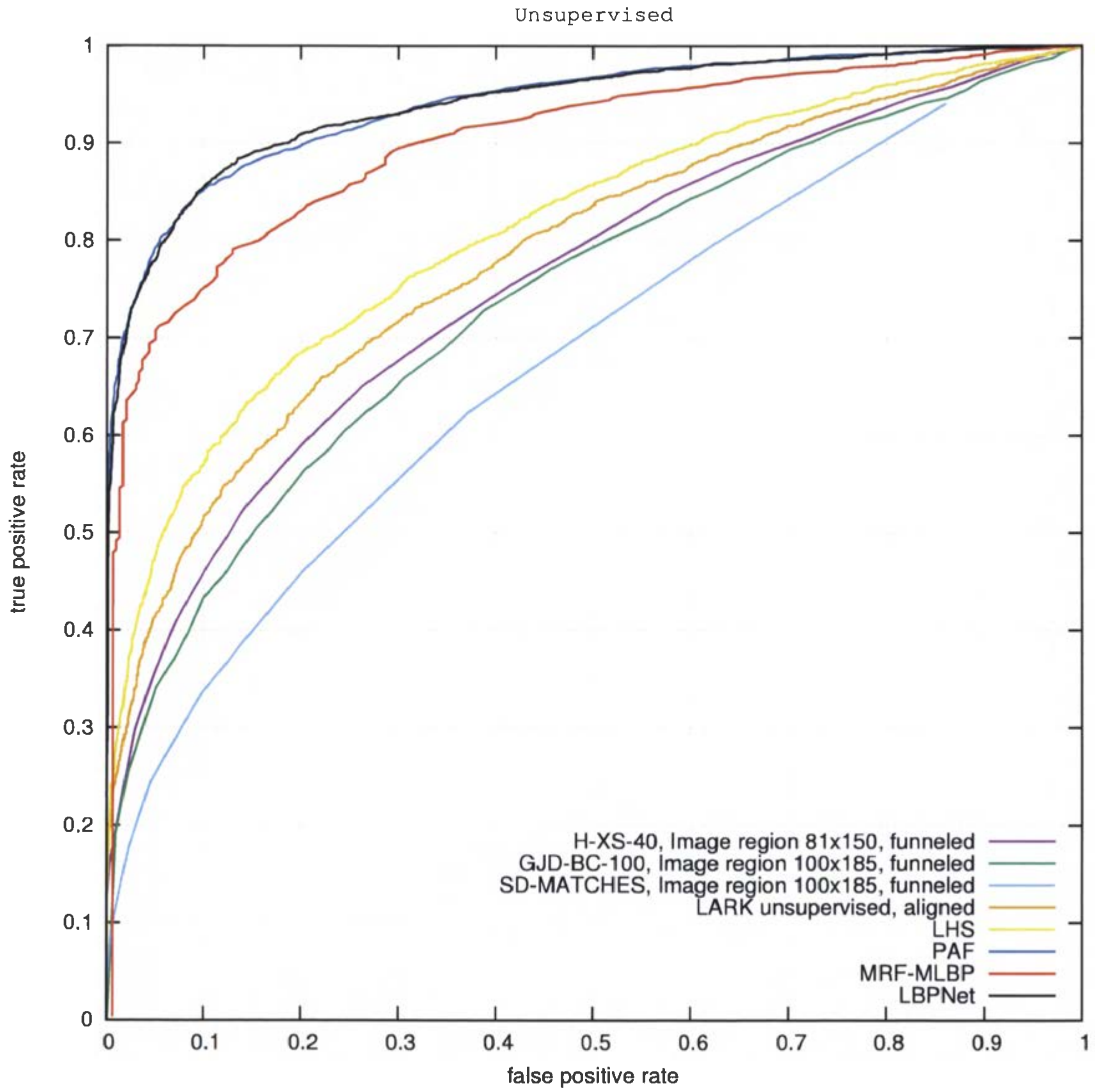


Figure 6.1: The ROC curves of various methods on LFW dataset view 2 with unsupervised setting

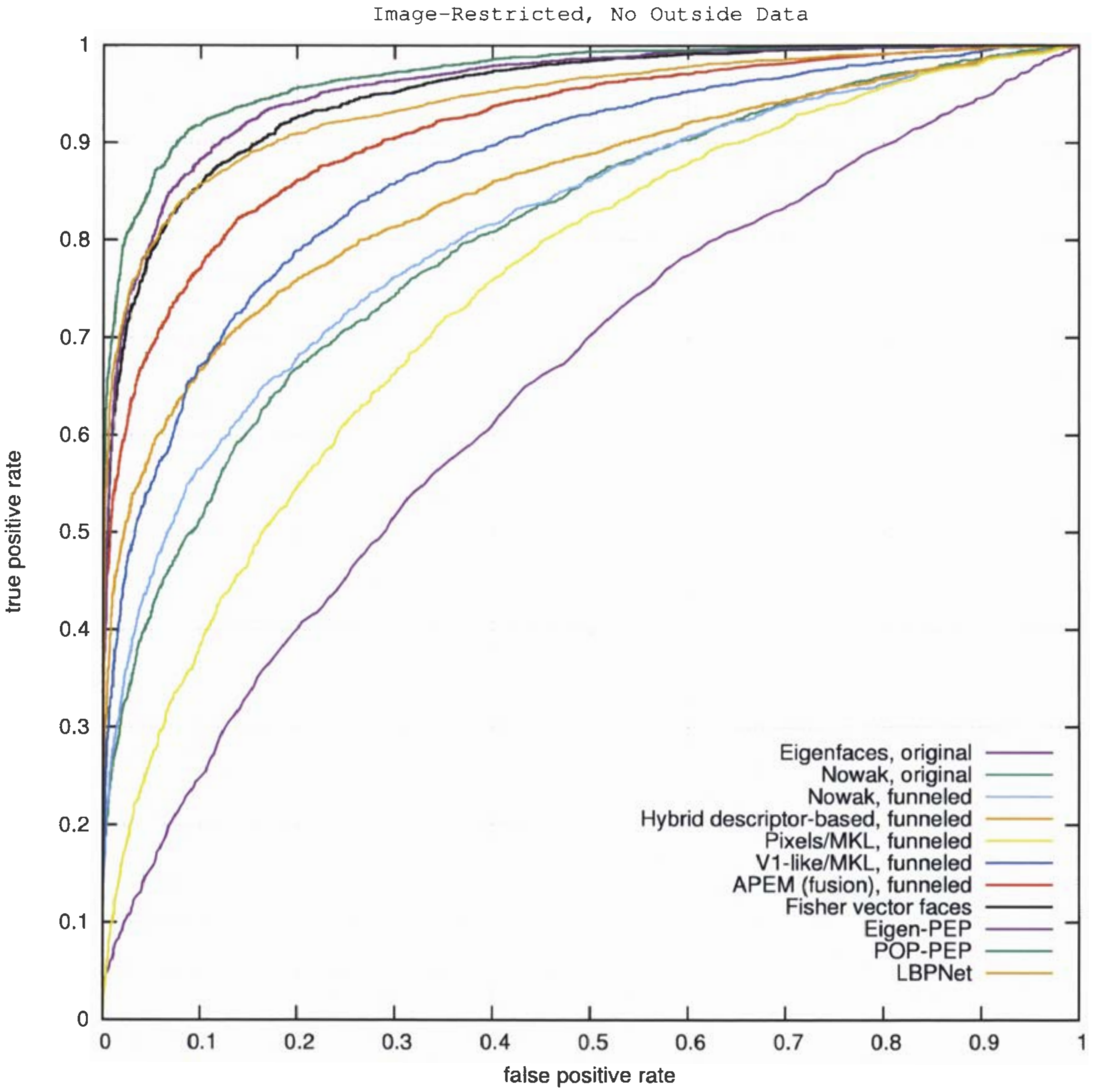


Figure 6.2: The ROC curves of various methods on LFW dataset view 2 with image-restricted no outside data setting

Table 6.3: Comparative results of various methods on LFW dataset view 2 with image-restricted no outside data setting

Methods	$\hat{\mu} \pm S_E$	Classifier
Nowak [59]	0.7393 ± 0.0049	
Hybrid descriptor-based [60]	0.7847 ± 0.0051	
3x3 Multi-Region Histograms (1024) [61]	0.7295 ± 0.0055	
Pixels/MKL [62]	0.6822 ± 0.0041	
V1-like/MKL [62]	0.7935 ± 0.0055	
MRF-MLBP [56]	0.7908 ± 0.0014	
APEM [33]	0.8408 ± 0.0120	
Fisher vector faces [36]	0.8747 ± 0.0149	Joint Bayesian
Spartans [58]	0.8755 ± 0.0021	CoMax-KCFA
Eigen-PEP [35]	0.8897 ± 0.0132	Joint Bayesian
POP-PEP [34]	0.9110 ± 0.0147	LDE
MRF-MBSIF-CSKDA † [63]	0.9363 ± 0.0127	CSKDA
LBPNet	$\mathbf{0.8772 \pm 0.0040}$	Cosine Similarity

†We only list the result which shows the best performance among the three reporting descriptors (i.e., MLBP, MLPQ, MBSIF) in their paper

Bayesian in [35]). Since in this experiment we just leverage the provided aligned dataset and employ simple NN classifier, the performance is compromised. Potential improvement is expected if we adopt similar strategies, but it is not our focus in this thesis.

6.3 Results on YTF

Table 6.4 reports our results on YTF comparing with other state-of-the-art methods. To fairly evaluate our method, we only select methods which do not require outside data for learning (same protocol as restrict setting on LFW). Our result is ranked second best in terms of both accuracy and AUC. When only comparing LBP based methods, our method outperforms all the other methods, especially improves the

Table 6.4: Comparative results of various methods on YTF with image-restricted no outside data setting

Methods	nm bm,gg mmn;lb	AUC	EER
Min dist, FPLBP [4]	65.6 ± 1.8	70.0	35.6
Min dist, LBP [4]	65.7 ± 1.7	70.7	35.2
$\|\|U1\prime U2\|\|$, FPLBP [4]	64.3 ± 1.6	69.4	35.8
$\|\|U1\prime U2\|\|$, LBP [4]	65.4 ± 2.0	69.8	36
MBGS L2 mean, FPLBP [4]	72.6 ± 2.0	80.1	27.7
MBGS L2 mean, LBP [4]	76.4 ± 1.8	82.6	25.3
MBGS+SVM- [64]	78.9 ± 1.9	86.9	21.2
APEM-FUSION [33]	79.1 ± 1.5	86.6	21.4
Eigen-PEP [35]	84.8 ± 1.4	92.6	15.5
LBPNet unsupervised	NA	**87.6**	**19.9**
LBPNet	$\mathbf{81.6 \pm 0.4}$	**88.1**	**20.0**

accuracy of LBP baseline method [4] about 15%.

We also report unsupervised results in Table 6.4. The insignificant difference between their AUC also implies that the performance is compromised because of the simple NN classifier.

6.4 Comparison With Baseline LBP methods

In addition to the deep learning architecture, LBPNet introduces two techniques based on the original LBP approach: (i) square root LBP descriptor, and (ii) WPCA that are used on extracted features. To further confirm the major improvement of LBPNet that comes from the adopted deep learning architecture, in this section we experimentally compare LBPNet to the variations of LBP method that have been created by combining LBP with the techniques reported in [28]. For ease of reference, we define these methods as baseline LBP methods.

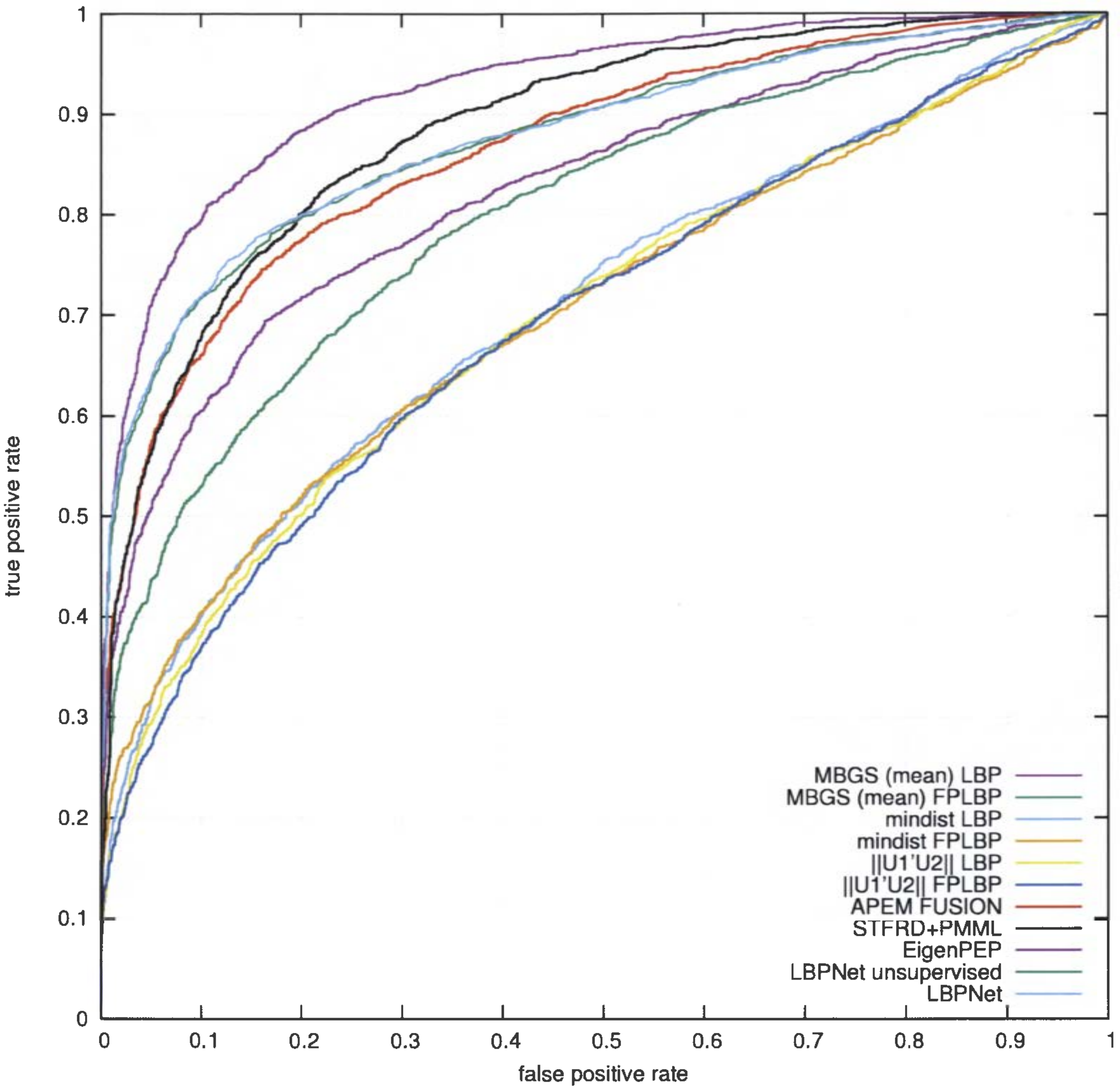


Figure 6.3: The ROC curves of various methods on YTF dataset

Regarding to the parameter settings, we use the operator $LBP_{2,8}^{u2}$ and cell size 10 for LBP and the select first 800 principal dimensions in PCA. Note that the LBP baseline result is different from [28] because different parameters and cropping region are used to keep consistency with the experiments in Chapter 5.

Table 6.5: Comparative results of baselines and LBPNet on FERET

Methods	Fb	Fc	Dup-I	Dup-II	Mean
LBP	0.966	0.974	0.706	0.684	0.878
sqrtLBP	0.976	0.974	0.755	0.735	0.900
sqrtLBP+WPCA	0.992	0.995	0.837	0.786	0.939
LBPNet	**0.996**	**1.00**	**0.942**	**0.936**	**0.978**

All the results on FERET are reported in Table 6.5. sqrtLBP denotes using the square root of the LBP operator and sqrtLBP+WPCA denotes using both the square root and WPCA of the extracted features. It can be concluded that LBP combining with these two techniques improve the performance considerately. sqrtLBP+WPCA outperforms LBP by 12% in terms of the mean accuracy. The LBPNet then further improve the accuracy by 4%. Especially in the most difficult probe sets Dup-I and Dup-II, LBPNet makes impressively improvement which are 10% and 14% respectively.

Table 6.6: Comparative results from LBPNet and baseline methods on LFW

Methods	AUC (Unsupervised)	Accuracy (Supervised)
LBP	0.7714	0.7088 ± 0.0058
sqrtLBP	0.7765	0.7108 ± 0.0041
sqrtLBP+WPCA	0.8849	0.7793 ± 0.0053
LBPNet	**0.9404**	**0.8772 ± 0.0044**

Not surprisingly, the results on LFW show the same pattern of achievement (Table 6.6). Square root LBP and WPCA greatly improve the performance of the original LBP approach, and the LBPNet further increases the accuracy by nearly 10%.

Table 6.7: Different predictions of baseline method (TT+sqrtLBP+WPCA) and LBP-Net on FERET

Probe set	Total Number	Different Predictions	Progression	Regression
Fb	1195	8	8	0
Fc	194	1	1	0
Dup-I	722	113	81	7
Dup-II	234	47	36	4
Fb+Fc+Dup-I	2111	122	90	7

Therefore, in both two cases the LBPNet brings significant improvements comparing with baselines. Considering all experiments are under the same settings, the only possible source of the improvement is the unique deep architecture in the LBPNet. In the rest of this section, we will further discuss the predictability and discrimination ability changes by study the distribution of their predictions.

6.4.1 Predictability

Although it has been demonstrated that LBPNet increase the overall recognition capability, it is still unclear that how it impacts to each individual class on the dataset. For ease of discussion, here we define: for every different prediction LBPNet makes, when it is a correct prediction, we call it a *progression*; otherwise, it is a *regression*. If the new method introduces not trivial number of regressions, it means this method at least is not superior than its baseline in some particular scenarios.

Table 6.7 shows predictability differences between LBPNet and its baseline. Some examples of different prediction by the baseline and the LBPNet are shown Figure 6.4. For probe set Fa and Fb which is recognized as easy to identify, the two methods predict same labels for most classes. All the different predictions on these two probe sets are progression. In the case of more challenged probe sets Dup-I and Dup-II, the LBPNet only gives 4 and 7 regressions. For the whole dataset, LBPNet gives regressed predictions on only 6% of samples. This result clearly shows that LBPNet improves the predictability of its baseline monotonously in almost all the conditions.

6.4.2 Discrimination ability

Here we define discrimination ability as how well the algorithm can cluster inputs into groups. In a perfect face recognition system, the (normalized) similarity of two images should be 1 when they are from the same person, and 0 for the different ones. However, this goal is unrealistic since someone (e.g., siblings) have more similar faces than others which will influence the similarity measurement. The similarity is thus always some value in between. The discrimination ability of a particular algorithm is measured by how significant the differences of the similarities are between different classified groups.

Two different algorithms with different discrimination abilities can give same one correct prediction for a particular dataset as shown in Figure 6.5. However, the one represented by Figure 6.5(a) is apparently worse than the other since it smears all the classes together. With only a small change on the classifier, it will fail on this classification task. In this section, the discrimination ability improvement of LBPNet will be discussed.

For the face identification system, we compare the subsets with the highest and second highest similarity (i.e., the first and second guess of the prediction). Higher distance between them indicates that this algorithm can better differentiate the target class from others. Table 6.8 presents the computed distances which is normalized by the max similarity. The table clearly shows that LBPNet better separate the images into two categories with normalized distance of 0.22 whereas it is only 0.06 and 0.19 in baselines, respectively. We also compare them only on their correct labelled classes. The results consistent with former ones with only about 0.005 increment for both three algorithms. It means that even for the subset of 100% correct labelled classes, LBPNet separates them with larger margin in the feature space.

Table 6.8: Mean normalized distance between the highest and second highest similarity from various algorithms on FERET

Methods	Distance	Distance On Correct Prediction Set
TT+sqrtLBP	0.0643	0.0722
TT+sqrtLBP+WPCA	0.1945	0.2092
LBPNet	0.2216	0.2268

For the face verification system, we carry out statistic analysis on the "matching pairs" and "mismatched pairs" set separately. The results are summarized in Table 6.9. The similarity are normalized by the max similarity of the whole dataset. The differences between the two means are about 0.09 in all these three compared algorithms. However, the standard deviations of the LBPNet are lower than the others. Specifically, the standard deviation of the matching set of the LBPNet is significantly lower than others. It means for the images from the same person, LBPNet gives constantly high similarity near to 1. This is well consistent with our intuitions in that people may rank the similarity level of different people in a large variance, but always give the same highest score for pictures from the same one.

Table 6.9: Distribution of the similarity computed from various algorithms on LFW

	Mismatched Pairs		Matching Pairs	
Methods	Mean	STD	Mean	STD
LBP	0.6079	0.0967	0.6975	0.0669
sqrtLBP+WPCA	0.8110	0.0889	0.9014	0.0304
LBPNet	0.8639	0.0841	0.9527	0.0168

Figure 6.6 shows the distributions of the baseline methods and the LBPNet. It

shows that although the mean distance keeps unchanged in these three methods, the discrimination ability varies because of the different of distribution. In LBPNet the curve of the matched set becomes tall and narrow which corresponds the lower standard deviation in Table 6.9. The curve of mismatched pairs set turns asymmetric with long tail in the opposite side of matching pair set. It also shows the increasing of discrimination ability even the standard deviation value does not reduce a lot.

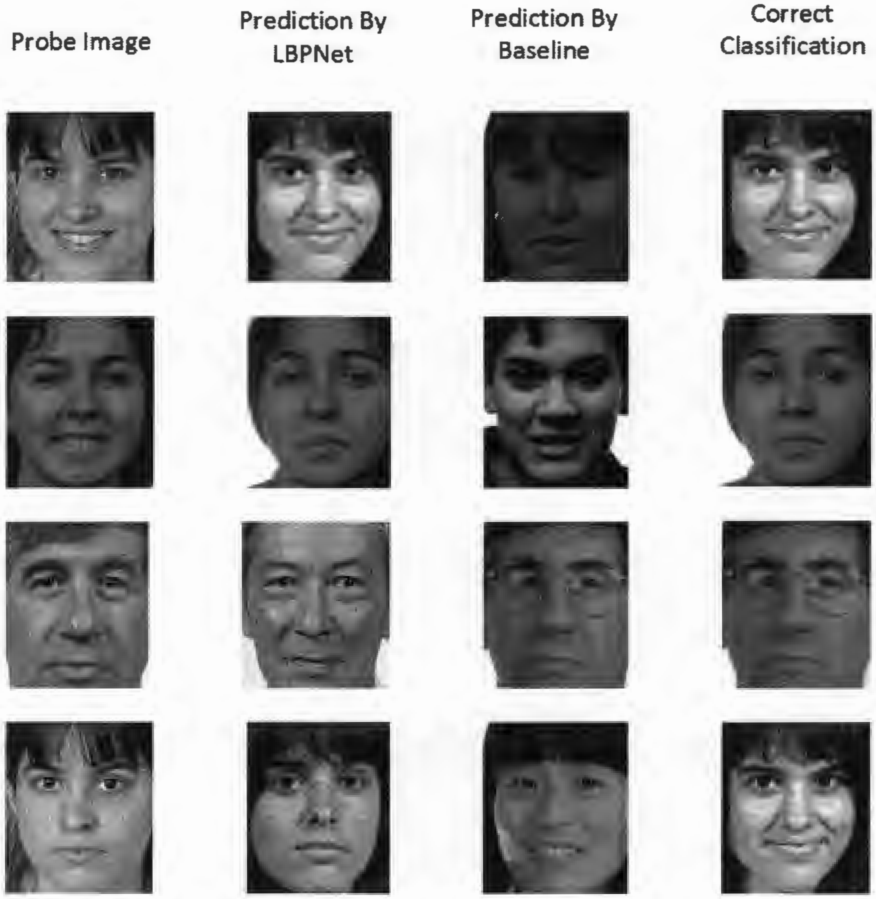
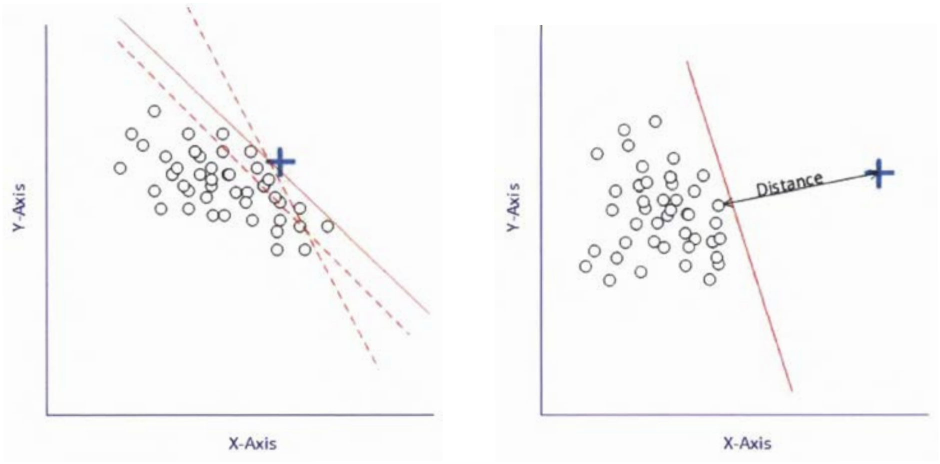


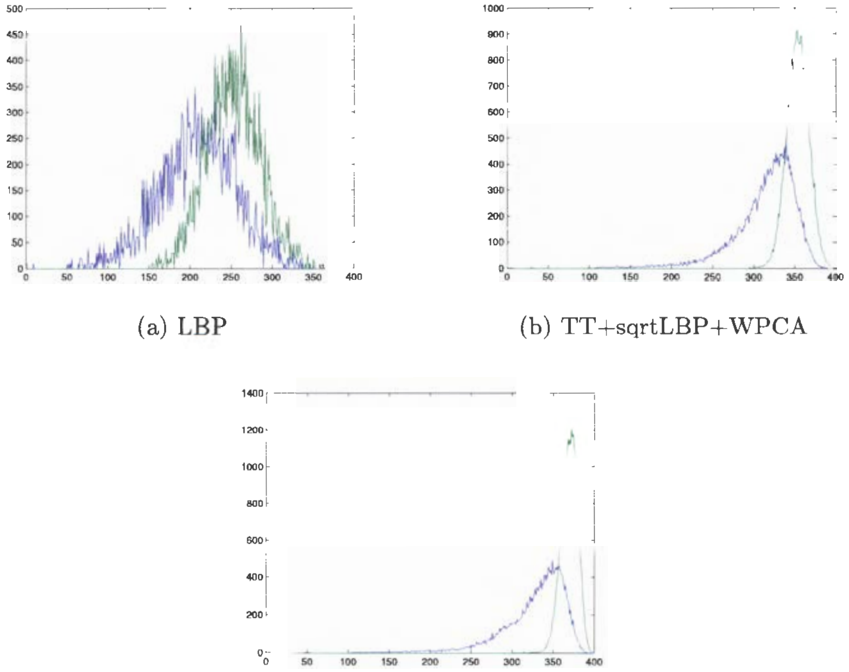
Figure 6.4: Examples of different predictions by baseline and LBPNet. From top to bottom, the first two are progressions and the third is regression. The last one none of them classify the subject correctly, thus it is neither progression nor regression.



(a) Classes are barely separated. Small changes of classier can lead to the fail of classification.

(b) Classes are separated with a big margin. The margin can be measured by the distance of samples from different class.

Figure 6.5: Example of algorithms with different discrimination ability shown in feature space. The classifier is represented by the red line.



(a) LBP

(b) TT+sqrtLBP+WPCA

(c) LBPNet

Figure 6.6: Distributions of baseline methods and LBPNet

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, a novel tool for face recognition named Local Binary Pattern Network (LBPNet) is proposed. This work is inspired by the successful LBP method and Convolutional Neural Network (CNN) deep learning architecture.

The LBPNet consists of two networks connected together: deep network part for feature extraction and simple network for classification. In feature extraction network, the discriminative representations are extracted progressively by two different filters: LBP filters and PCA filters. LBP filters are based on LBP descriptors described in [19], while PCA filters reduce feature dimensionality by feature selection and subspace projection. All the two filters are replicated densely on the input maps. In each layers, filters with different parameters are employed to capture multi-scale statistic. The extracted features in LBPNet are: (i) high-level features extracted from low-level LBP features, which are more robust to variability; (ii) over-complete features which contains redundant information from overlapped filters and multi-scale analysis. The classification network is based on a simple nearest neighbourhood classifier. By connecting to two feature extraction networks, two sets of features from two different images are accepted, and the overall similarity are computed hierarchically in this part.

Extensive experiments were conducted on several public benchmarks (i.e., FERET, LFW and YTF) to evaluate our method. LBPNet achieves promising results comparing to the other methods in the same category: its results outperforms (on FERET) or is comparable (on LFW and FERET) to other methods in the same categories, which are single descriptor based unsupervised learning methods on FERET and LFW, and single descriptor based supervised learning methods with image-restricted no outside data settings on LFW and YTF, respectively. We also conducted experiments between LBPNet and the baseline LBP methods. The baseline LBP methods are defined as the original LBP method or it combining with one or more techniques used in LBPNet. The results showed that LBPNet improves the baselines fundamentally in terms of both predictability and discrimination ability.

Comparing with CNN, the LBPNet retains a similar topology: (i) the network employs multiple processing layers to gradually extract features; (ii) features are extracted in a heavily overlapped manner; (iii) layers are partially connected to simplify the model; (iv) multiple kernels are used in one lay to obtain multi-scale representations. The most significant architectural difference between LBPNet and CNN is that LBPNet uses off-the-shelf computer vision descriptor (LBP descriptor, in our case) instead of trainable convolution kernel in CNN. Comparing with the regular LBP and CNN method, the LBPNet has the following achievements:

High Accuracy Comparing with the original shallow LBP method, LBPNet, which is a deep learning architecture based LBP method, improves the recognition accuracy significantly. The extensive experiments show that the main improvements come from its deep learning architecture.

No costly learning approach required Two layers in LBPNet are learn-based layer: PCA filter layer (supervised and unsupervised learning) and output layer (supervised learning only). Comparing with regular CNN method whose learning is based on back-propagation and gradient decent, the learning approaches on these two layers are simple and fast.

Comparing with the regular CNN method, LBPNet avoids the computation expensive learning approach. LBPNet only require a simple learning stage in PCA filter layer and output layer (supervised learning only). The

No additional training data required The regular CNN needs to be learned on massive training data due to the large number of parameters it contains, while the LBPNet can be learned on a relatively small training set. As a result, the LBPNet do not require outside data to train its model even on the challenging LFW and YTF datasets.

7.2 Future Work

Some of many areas which are potential to continue to enhance our proposed framework are summarized as follows.

- Although LBPNet is proposed based on LBP descriptor, this deep learning network is not limited to it. One possible alternative is SIFT descriptor [20] which is utilized by many state-of-the-art systems [36, 35, 34].

- The dimensionality can be reduced discriminately with supervised learning algorithm (e.g., LDA) instead of PCA in the second layer.
- It is possible to fold more processing layers in deep network part of LBPNet. Comparing with regular CNN which usually contains three or more layers, the abstraction level of extracted features in LBPNet may not be as high as in CNN.
- The results on LFW and YTF under supervised learning protocol (Table 6.3 and Table 6.4) suggest that, superior results can be achieved with sophisticated classifiers. Replacing the simple NN classifier in LBPNet with other learn-based method (e.g. Joint Bayesian [65]) could improve performance in this category.

Bibliography

[1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[2] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[3] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1, pp. 947–954, IEEE, 2005.

[4] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 529–534, IEEE, 2011.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] S. Z. Li and A. K. Jain, "Handbook of face recognition," 2011.

[7] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *arXiv preprint arXiv:1503.03832*, 2015.

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

[11] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *arXiv preprint arXiv:1509.00244*, 2015.

[12] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *arXiv preprint arXiv:1412.1265*, 2014.

[13] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1891–1898, IEEE, 2014.

[14] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, pp. 1988–1996, 2014.

[15] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," *arXiv preprint arXiv:1404.3543*, 2014.

[16] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153, IEEE, 2009.

[17] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3642–3649, IEEE, 2012.

[18] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

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

[20] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.

[21] L. Shen and L. Bai, "A review on gabor wavelets for face recognition," *Pattern analysis and applications*, vol. 9, no. 2-3, pp. 273–292, 2006.

[22] C.-H. Chan, J. Kittler, and K. Messer, *Multi-scale local binary pattern histograms for face recognition.* Springer, 2007.

[23] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3025–3032, IEEE, 2013.

[24] S. U. Hussain, T. Napoléon, and F. Jurie, "Face recognition using local quantized patterns," in *British Machive Vision Conference*, pp. 11–pages, 2012.

[25] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3554–3561, IEEE, 2013.

[26] X. Tan and B. Triggs, "Fusing gabor and lbp feature sets for kernel-based face recognition," *Analysis and Modeling of Faces and Gestures*, pp. 235–249, 2007.

[27] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[28] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Computer Vision–ACCV 2010*, pp. 709–720, Springer, 2011.

[29] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2707–2714, IEEE, 2010.

[30] V. Perlibakas, "Distance measures for pca-based face recognition," *Pattern Recognition Letters*, vol. 25, no. 6, pp. 711–724, 2004.

[31] N.-S. Vu and A. Caplier, "Enhanced patterns of oriented edge magnitudes for face recognition and image matching," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 1352–1365, 2012.

[32] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 1. Wiley New York, 1998.

[33] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3499–3506, IEEE, 2013.

[34] H. Li and G. Hua, "Hierarchical-pep model for real-world face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4055–4064, 2015.

[35] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-pep for video face recognition," in *Computer Vision–ACCV 2014*, pp. 17–33, Springer, 2015.

[36] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," in *British Machine Vision Conference*, 2013.

[37] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "Plbp: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, no. 10, pp. 2502–2515, 2011.

[38] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 786–791, IEEE, 2005.

[39] T. Jaakkola, D. Haussler, *et al.*, "Exploiting generative models in discriminative classifiers," *Advances in neural information processing systems*, pp. 487–493, 1999.

[40] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*, pp. 143–156, Springer, 2010.

[41] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[42] L. Chen and N. Tokuda, "A unified framework for improving the accuracy of all holistic face identification algorithms," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 107–122, 2010.

[43] L. Chen, L. Yan, Y. Liu, L. Gao, and X. Zhang, "Displacement template with divide-&-conquer algorithm for significantly improving descriptor based face recognition approaches," in *Computer Vision–ECCV 2012*, pp. 214–227, Springer, 2012.

[44] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[45] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *null*, p. 958, IEEE, 2003.

[46] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1875–1882, IEEE, 2014.

[47] P. J. Phillips, H. Moon, S. Rizvi, P. J. Rauss, *et al.*, "The feret evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 10, pp. 1090–1104, 2000.

[48] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1635–1650, 2010.

[49] S. Xie, S. Shan, X. Chen, and J. Chen, "Fusing local patterns of gabor magnitude and phase for face recognition," *Image Processing, IEEE Transactions on*, vol. 19, no. 5, pp. 1349–1361, 2010.

[50] N.-S. Vu, "Exploring patterns of gradient orientations and magnitudes for face recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 2, pp. 295–304, 2013.

[51] Z. Chai, Z. Sun, H. Mendez-Vazquez, R. He, and T. Tan, "Gabor ordinal measures for face recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 1, pp. 14–26, 2014.

[52] H. V. Nguyen, L. Bai, and L. Shen, "Local gabor binary pattern whitened pca: A novel approach for face recognition from single image per person," in *Advances in Biometrics*, pp. 269–278, Springer, 2009.

[53] L. Yan, "Multi-scale local binary pattern histograms for face recognition," Master's thesis, 2013.

[54] J. Ruiz-del Solar, R. Verschae, and M. Correa, "Recognition of faces in unconstrained environments: a comparative study," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 1, 2009.

[55] H. J. Seo and P. Milanfar, "Face verification using the lark representation," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 4, pp. 1275–1286, 2011.

[56] S. R. Arashloo and J. Kittler, "Efficient processing of mrfs for unconstrained-pose face recognition," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pp. 1–8, IEEE, 2013.

[57] D. Yi, Z. Lei, and S. Z. Li, "Towards pose robust face recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3539–3545, IEEE, 2013.

[58] F. Juefei-Xu, K. Luu, and M. Savvides, "Spartans: Single-sample periocular-based alignment-robust recognition technique applied to non-frontal scenarios," 2015.

[59] E. Nowak and F. Jurie, "Learning visual similarity measures for comparing never seen objects," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[60] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[61] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics*, pp. 199–208, Springer, 2009.

[62] N. Pinto, J. J. DiCarlo, and D. D. Cox, "How far can you get with a modern face recognition test set using only simple features?," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2591–2598, IEEE, 2009.

[63] S. R. Arashloo and J. Kittler, "Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 12, pp. 2100–2109, 2014.

[64] L. Wolf and N. Levy, "The svm-minus similarity score for video face recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3523–3530, IEEE, 2013.

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