Exploring suitable Classifier for robust Face and Facial Expression Recognition

by

Deshna Jain
201111003

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

July, 2013

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Deshna Jain

Certificate

This is to certify that the thesis work entitled Exploring suitable Classifier for robust Face and Facial Expression Recognition has been carried out by Deshna Jain for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Dr. Suman K. Mitra
Thesis Supervisor

Insert

Acknowledgments

At first, I owe my deepest gratitude to Prof. Suman K. Mitra, whose guidance, encouragement and positive criticism enabled me to become familiar with the subject. It would have been merely a dream to finish this thesis work without his continuous support and motivation.

I am also thankful to my thesis examiner, Prof. Aditya Tatu from DA-IICT, Gandhinagar, for their valuable suggestions on my work. Besides them, I would like to express my gratitude to the authors of all the books and publications, which helped me in enhancing my knowledge and understanding about my work.

I am indebted to my family for their trust in me that made me confident enough to accomplish my work. Their support in a number of ways must also be acknowledged here. Even at times when completion of thesis on time seemed unlikely their unfazed trust helped me through.

I would like to thank my friends for their continuous support trust and encouragement. I owe a special thanks to Gitam Shikkenawis and Ashish Phophalia for their prompt help and support throughout my work.

Contents

Abstract

Face recognition by machines has been studied since last few decades and the problem is attempted to be solved by various ways. However any robust solution is not acheived yet by the researchers due to numerous challenges involved like illumination changes, pose variation, occlusion, cluttered background, non-cooperation of the subject and ageing effect on human face. We have worked by modelling the problem as a pattern recogniton problem. The solution of this problem involves mainly three steps: (a) Face detection and segmentation, (b) Feature extraction, and (c) Classification or recognition. We have worked on finding the robust classifier for face and facial expression recognition.

Naive Bayes Classifier (NBC) is the statistical classifier that works by estimating the maximum probability of the possible classes to which the testing data point may belong assuming that the features are mutually independent. It makes use of Bayes rule for likelihood computation. This approach works well if the distribution of the features is known accurately. Otherwise, probability distribution of the features belonging to corresponding classes has to be estimated with density estimation techniques. Here features are assumed to follow Gaussian distribution. Experiments are done for classifying faces from YALE face database and DAIICT database, taking ELPP coefficients as the features. Another classifier we used is Support Vector Machine (SVM) that works by finding the decision plane between two classes. It finds the decision plane with the help of support vectors having maximum margin between them. Experiments performed with SVM give better results than NBC for both DAIICT and YALE face database.

While using NBC, one of the estimation techniques that is used in this work is Kernel Density Estimation also known as Parzen window. The approach estimates the density of a point for a given dataset with a global bandwidth. This classification technique is used for face recognition using YALE face database and DAIICT database. For DAIICT database the estimation method shows different results for the same dataset with different parameters whereas no significant re-

sults are obtained for YALE face database. On the other hand, in the whole algorithm there is no measure of best fit of the estimatd curve involved. These issues are resolved by using Pearson's chi-squared test for testing goodness of fit of the estimation with changing parameters of the selected bandwidth. In addition to this, bandwidth is kept dynamic by computing it with neighboring datapoints instead of keeping it global. This approach performs better than the former one for YALE face database and equivalent for DAIICT database.

The experiments are extended for classifying the facial expressions as well. A comparision of KNN, NBC, proposed approach for NBC and SVM is presented in the work. SVM outperformed all the classifiers for both the databases.

List of Tables

List of Figures

CHAPTER 1
Introduction

Face Recognition is one of the most popular research problem since 30 years. The reasons for which it is of great significance are the need of the reliable biometric system for various applications. Today in this digital era, we have most of our assets secured by computers by incuring different security mechanisms such as finger prints, passwords, encryption keys. But intruders are apparantly successful at breaching the security systems by just cracking the keys and passwords and faltering the finger prints. So, we need a robust system to seize the unauthorized access. One of the prospective solution is availing the biometric information which is unique to every person and can't be counterfeited. Human face is the strong unique characterstic to make use of. Other various practical applications of a Face Recognition(FR) System are forensic identification, human computer interaction, video surveillance and law enforcement.

Human beings have the natural ability of recognizing human faces and distinguishing them from each other. The phenomena of the human face perception is still under research and involves many factors. Human visual system is as good as possible at recognizing faces and withstand all the challenges involved in it such as varying distance from the subject, it's age, illumination, poses, expressions, occlusions and many more. So, it is imperial to understand the human visual system and take the cues by which such a system can be built. Machines have advanced in detecting and recognizing faces to some extent which needs constrained environment and cooperation of the subject. Yet machines have not attained the required proficiency in doing so. The requirement of biometric personal identification has risen with time.

Following sections in this chapter give a brief about what a *Face Recognition* problem is, and how it can be modeled as a Pattern Recognition problem. Then different ways of solving a Pattern Recognition problem are discussed and a liter-

ature survey of the Face Recognition is presented.

1.1 *Face Recognition* Problem

Face Recognition is the task which involves the comparison of a given facial image with a set of various images of different persons with known identies and identify the person to which the given image belongs. The given face image is called the *Test Image* and the set of images with known identities is called the *Training Data*. The method to solve the problem may involve some preprocessing of the images like denoising and then detection and segmentation of the faces. There are many challenges imposed in building a FR system which mainly consist of pose variation, different illumination, uncontrolled environment and occlusion. The systems so far built work well under constrained conditions.

1.2 How FR is a PR problem

Face Recognition problem can be viewed in various aspects like a computer vision problem, image processing problem, signal processing problem or a pattern recognition problem. Here this problem is modeled as a pattern recognition problem. A pattern recognition problem typically attempts to find the class or the category to which an input pattern may belong. This task is accomplished by either taking advantage of the known class labels of some given patterns or by finding the inherent clustering present in the dataset.

Here a facial image is represented in the form of feature vector obtained by applying a tranformation on the facial image. This feature vector is then treated as a pattern for further classification.

1.3 Ways to solve PR problem

A Pattern Recognition problem can be solved by different approaches. Earlier attempts were based on *template matching* where the input pattern is treated as a template and matched with all the prestored templates belonging to predefined classes [3]. Another approach is *syntactic* in nature, where a pattern is assumed to be a combination of subpatterns and these subpatterns are again made up of simpler subpatterns. The simplest subpatterns are called *primitives* and the complex patterns are believed to follow some structure of the various combinations of

these primitives. This is analogous to a language made up of alphabets following a particular grammer. Thus, a collection of patterns is viewed as a language made up of few primitives following a set of rules of construction. These rules are inferred from the training data for each class. An input pattern is then checked for the class of which rules are followed. Third approach is *statistical* based [3], where each pattern is represented in the form of a d dimensional vector. These dimensions are chosen such that the patterns belonging to different classes are projected away from each other so that they can be easily classified. A decision boundary is formed between the classes that separates them by some discriminant function. Subsequently, an input pattern is projected in the same d dimensional space and classified on the basis of its location with respect to predefined classes.

1.4 State of the art of Face Recognition

Applications of Face Recognition Techniques take the input ranging from static and specific format images to uncontrolled video images. This variety of inputs poses many technical challenges and requires plenty of image processing and pattern recognition techniques. FRT systems can be classified in two types: One those work on static, controlled images and another working with video. These two categories differ in various aspects like image quality, background clutter, occlusions, range of images of an individual and the user input.

The face recognition problem can be stated as identifying the individual to which the given input facial image belongs by comparing it to a database of images with known identities. Note that input image can be obtained from a photograph or from a video by applying some preprocessing for noise suppression and segmentation of the facial image. The whole solution of the problem involves a multi step procedure which is discussed later in this section. Research on face recognition by machines had started in 1970s [4] and by the influential work of Kanade [5]. The problem of face recognition has attracted researchers from various fields spanning psychophysicists, neuroscientists, and engineers of image processing, pattern recognition, neural networks, computer vision, and computer graphics.

The problem of automatic face recognition from the still images comprises three main subtasks: (1) detection and rough normalization of faces, (2) feature

extraction and accurate normalization of faces, (3) identification and/or verification. These three subtasks are critical in the face recognition automation and may not be totally separated. But isolating them make the process more enhanced and easier to resolve the problem with various state of the art methods dedicated to every task. In the following sections we give a summary of the recent methods incorporated in the subtasks.

1.4.1 Segmentation/Detection

The earlier research work done in 1990s on Face detection mainly focused on the single face segmentation from the cluttered background. These approaches mainly used a deformable feature-based template, a whole-face template, skin color, and a learning algorithm. Among various techniques, appearance and image-based method [6] [7] give best results. These methods work with the similar looking features of faces which distinguish them from non face images. Training the computer with a large number of samples of facial images works successfully. Recently, approach based on training on multiple-view samples [8] [9] has been studied. Multiview-based methods of face detection and recognition appears to work better than invariant feature-based method. Treating face detection as a two-class classification problem helps to reduce false positives dramatically [6] [7] while maintaining true positives.

1.4.2 Feature Extraction

Many face recognition systems work with facial features instead of working with whole facial image. Holistic matching methods like Fisherfaces [10] and eigenfaces [11], need accurate locations of key facial features such as nose, eyes, and mouth to normalize the detected face [12] [13]. Three types of feature extraction methods can be distinguished: (1) generic methods based on edges, lines, and curves; (2) feature-template-based methods that are used to detect facial features such as eyes; (3) structural matching methods that take into consideration geometrical constraints on the features. Approaches focusing on individual features [14] used to fail with a significant change in the appearance of face like openmouth, eyes with glasses and closed eyes. Recent approaches incorporate structural matching methods, like Active Shape Model [15] for reliable feature detection.

A template-based approach for eyes and mouth detection is presented in [16].

This approach uses two predefined parameterized templates of eyes and mouth to match with the face regions. Matching is done by linking the peaks, edges and valleys in the image to the corresponding properties in the template with the help of an energy function. By changing the parameters of this function iteratively, this function can be minimized to fit the image. On the other hand, the statistical shape model presented in [15] provides more robustness and flexibility. To overcome variation in texture, the ASM model is extended to statistical appearance models including a Flexible Appearance Model (FAM) [17] and an Active Appearance Model (AAM) [18].

1.4.3 Recognition from Intensity Images

Face recognition uses a large number of techniques motivated by different principle. A broad classification of the approaches is as given under:

Holistic matching methods

These methods use the whole face region as the raw input to a recognition system. Many methods that come under this category are: PCA, Eigenfaces, Probabilistic eigenfaces, Fisherfaces/subspace LDA, SVM, ICA and LDA/FLD and Probabilistic decision based NN (PDBNN). *Principle Component Analysis* (PCA) projections [19] is a compact representation of the normalized face images by reducing the redundancies existing among them. It works on the principle that the sample vectors x can be represented as the linear combinations of the orthogonal basis Φ_i: $x = \sum_{i=1}^{n} a_i \Phi_i \approx \sum_{i=1}^{m} a_i \Phi_i$ ($m \ll n$) by solving the eigenproblem

$$C\Phi = \Phi\Lambda$$

where C is the covariance matrix for input x.

Such representations are more robust against noise, blurring, changes in background and partial occlusion. First use of eigenfaces was presented in [11] for face detection and identification. Every face in the database is represented as a vector of weights, given the eigenfaces. The weights are computed by projecting the image into the eigenface components. For the test image, the weights are obtained similarly and are matched with the image having approximately same weights.

This approach is extended by using probabilistic measure for similarity of eigenfaces, instead of Euclidean distance. It was proposed by [20] with the Bayesian

analysis of image difference. This approach solves a two-class problem rather than multiclass problem. The two classes defined are: Ω_I, representing intrapersonal variations between multiple images of the same individual, and Ω_E representing extrapersonal variations due to differences in identity. Both classes are assumed to follow Gaussian distribution and for a given intensity difference, $\Delta = I_1 - I_2$, likelihood functions $P(\Delta|\Omega_I)$ and $P(\Delta|\Omega_E)$ are computed. Using MAP rule and the likelihood functions, two face images are matched if $P(\Delta|\Omega_I) > P(\Delta|\Omega_E)$. This approach reported remarkable imrovement in the performance over standard nearest-neighbor eigenspace matching [21].

Another approach using LDA/FLD have been presented in [10]. It uses scatter matrix analysis for LDA training. The within and between-class scatter matrices S_w and S_b respectively, are computed as:

$$S_w = \sum_{i=1}^{M} Pr(\omega_i) C_i,$$

$$S_b = \sum_{i=1}^{M} Pr(\omega_i)(m_i - m_0)(m_i - m_0)^T$$

where $Pr(\omega_i)$ is the prior probability of the class. Here C_i in the scatter matrix S_w denotes the average scatter of the sample vectors x belonging to different classes ω_i around their means m_i: $C_i = E[(x(\omega) - m_i)(x(\omega) - m_i)^T|\omega = \omega_i]$, whereas S_b represents the scatter of the conditional mean vectors m_i around the overall mean vector m_0. These matrices are used for quantifying the discrimating power as: $J(T) = |T^T S_w T|/|T^T S_w T|$. The optimal projection matrix W which maximizes $J(T)$ can be obtained by solving a generalized eigenvalue problem:

$$S_b W = S_w W \Lambda_W$$

The similarity lying in all the projection methods is that classification is done by projecting the input x into a subspace via a projection/basis P_{roj} :

$$z = P_{roj} x$$

and then finding the closest projection vectors in prestored labeled classes to the input projection vectors. Different algorithms of finding projections and different measures of similarity account for over all performance of the system [22]. Another kind of application of using these methods can be found in [23] where

a recursive training of eigenspace and LDA projections is performed in a tree-structure of labeled classes. This method first broadly classifies face and non-face categories and then classifies face images to which class of individual they belong. A comparision and analysis of performance is shown in [24]. The candidate methods of comparision are (a) an eigenface method, (b) a correlation-based method, (c) a variant of the linear subspace method, and (d) a Fisherface method which uses subspace projection prior to LDA projection to avoid the possible singularity in S_w as in [23]. Experimentally, it was found that Fisherman method obtains significant better results than other three methods. Further enhancement of LDA-based systems was carried out by [25]. They used weighted distance measure for finding close projection vectors and uniquely select the global subspace dimension, as well as a regularized procedure for modifying the within-class scatter matrix is used.

Keeping the heuristic that high-order statistics can be more informative as the key idea, Independant Component Aanalysis (ICA) was prosposed in [26] for feature extraction. ICA decorrelates second-order moments as well as high-order moments of the input. PCA is performed first to reduce the dimensionality of the input images. Then obtained eigenvectors are used as input for performing ICA. This methodolgy was followed in [27] where statistically independent source images are considered as independent image features for a training image set. In [28] image filters are used to produce statistically independent outputs and then ICA performed on some highest significant eigenvalues obtained from the PCA.

Another approach based on probabilistic decision-based neural network (PDBNN) or face detection and recognition involves neural network in [29]. The proposed system first detects face, then localizes eye region and then recognizes face. Localizing only the eye region gives robustness to system against expression variation. Modular structure of the system is an advantage for PBDNN, so that a single subnet is dedicated to a single class. This gives faster convergence rate and less number of hidden layers in the neural network of the system in comparison to Multilayer Perceptron. However, large number of classes may create problems for the sytem.

Feature-Based Structural Matching Approaches

These methods are based on local features such as eyes, nose and mouth. These features are first extracted and their local statistics are used by structural classi-

fier. Early methods of structural matching were based on local features geometry, 1D and pseudo-2D HMM algorithms [30]. Elastic Bunch Graph Matching (EBGM) [31], [32] was a successful system based on Dynamic Link Architecture (DLA) [33]. This system obtains image graphs by extracting important facial features and represent them by wavelet components (jets) as in [31]. Wavelet representation is robust to rotation, distortion and illumination variations. DLAs in [33] use two parameters, one is the J_{ij} and another is T_{ij}. J-variables act as synaptic weights of the connection of neuron$i$ and neuron j and are dynamic in nature whereas T-parameters are used as contraints for J-variables. T-parameters change gradually with long-run synaptic plasticity whereas J-variables change rapidly. Model graph is used for the representation of the stored image, and these model graphs serve as the pattern classes. A new input image is matched by forming the image graph and comparing it with all the prestored model graphs.

An extension of this approach is Elastic Bunch Graph Matching (EBGM) [32]. This approach differs from DLAs in using bunch graph representation for a class instead of model graph. This helps in confronting the pose variations of the face.

Hybrid Approaches

Inspired by human perception system, these approaches use both holistic methods and local features matching methods for face recognition. These approaches are believed to perform better as they inherit the advantages of both the approaches. Different methods categorized under it are: Modular eigenfaces, Hybrid Local Feature Analysis (LFA), Shape-normalized and Component-based. A face recognition system can be made more capable by using multiple views of a person. These different views have separate eigenspaces and set of images of each view has its own eigenspace. This approach is known as *view-based eigenspaces*, and found to perform better than the approach that uses eigenfaces of all the images of all views. Another method in [34] with enhanced performance takes eigenfeatures into account along with eigenface.

It has been shown earlier that natural images contain some redundancies [35] and when a set of normalized face images is considered , these redundancies tend to increase significantly. So, these images can be expected to be represented in lower dimensions. LFA takes the advantage of this redundancy and extract topographic local features from the global PCA modes. An appearance based method is proposed in [17]. This approach uses shape as a feature in addition to the in-

tensities of the face images. System is trained with different shapes of the images and tests the input image by deforming its shape in different ways for different classes. Another method which is component-based is presented in [36]. This method uses decomposition of face into a set of facial components that are interconnected by a flexible geometrical model.

CHAPTER 2
Feature Extraction

A Face Recognition system is trained using a set of preprocessed and labelled images which is then used to predict the class for the unlabelled images. An 256X256 image in natural form if projected to the existing feature spaces found in the literature, would be of 65536 features. But, natural images are highly correlated and a redundancy exists in their representation and hence they can be uniquely represented in a lower dimensional subspace. Lower dimensionality incurs lower computational cost and no overfitting of the training data.

The lower dimensional manifold in which the high dimensional data may be embedded can be linear or nonlinear in nature. Various linear and nonlinear dimensionality reduction techniques have been proposed yet. Linear dimensionality reduction techniques mainly include Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA) whereas nonlinear dimensionality reduction techniques are Isometric Feature Map (ISOMAP), Locally Linear Embedding (LLE), Laplacian Eigenmaps, Locality Preserving Projection (LPP), Extended LPP (ELPP) and Extended supervised LPP (ESLPP). After applying any of these relevant techniques to the facial images, significant features are extracted and used for further training and testing process of the system.

2.1 Locality Preserving Projection (LPP)

LPP [37] is a linear dimensionality reduction technique used for nonlinear data manifold, which preserves the local information present in the data in high dimensional subspace. This neighborhood information is stored using a graph, which is further used for finding transformation matrix. This transformation projects the data points in such a way that neighboring points remain as close as possible. Locality Preserving Projections are obtained by finding the optimal

linear approximations to the eigenfunctions of the Laplace Beltrami operator on the lower dimensional manifold. This preserved neighborhood characteristic of the data is vital for finding the clustering of the data or its classification in different classes.

The notion of LPP considers every data point as a node. Above these nodes, a graph is built by connecting the nodes by edges. These edges are assigned weights by a weight function. The edge connecting distant nodes is assigned lower weight as compared to the one connecting neighboring nodes. This graph is represented by an adjacency matrix S.
The algorithm is as given below: The problem can be stated as, given a set of l data points $X = \{x_1, x_2, \ldots, x_l\}$ each of dimension n i.e. $x_i \in \mathbf{R}$, a transformation matrix A is to be found such that $Y = AX$, where $Y = \{y_1, y_2, \ldots, y_l\}$ of dimension m and $m << n$.

- **Adjacency matrix construction:** Let G be a graph with l nodes corresponding to the data points, in which two nodes i and j are connected by an edge if their corresponding nodes x_i and y_i are neighbors. The closeness is measured by following two notions:
  - ϵ **- neighborhood:** Points x_i and y_j are neighbors if $||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon$, where the norm is simple Euclidean norm in $\mathbf{R}^n$ and $\epsilon \in \mathbf{R}$.
  - k **- neighborhood:** Points x_i and y_i are neighbors if x_i is k-nearest neighbor of x_j or x_j is k-nearest neighbor of x_i.
- **Assigning Weights:** There are two varied methods of assigning weights to an edge connecting nodes i and j:
  1. **Heat kernel:** The weight is assigned as:
  $$W_{ij} = e^{-\frac{||x_i - x_j||^2}{t}} \tag{2.1}$$
  where $t \in \mathbf{R}$.
  2. **Simple minded**: $W_{ij} = 1$ if nodes i and j are connected.
- **Objective Function:** The transformation vector W that minimizes the objective function is given by the solution to the generalized eigenvalues problem: $XLX^TW = \lambda XDX^TW$

  where, $L = D - S$ is the Laplacian matrix, $D_{ii} = \sum_i S_{ij}$, XLX^T and XDX^T are symmetric and positive semi definite.

2.2 Extended Locality Preserving Projection (ELPP)

ELPP [38] is extension of LPP that attempts to eradicate the ambiguity occuring in the overlapping region of classes and increases the rudicibility power as well. It incorporates a different weighing function which gives significance to points which are moderatly apart along with the nearest points.

The notion of formulation of ELPP is taking the advantage of points which are at a moderate distance from the point of interest, in addition to the nearest neighbors in preserving its local information in the adjacency graph. In LPP, for constructing adjacency matrix, K-NN is used for determining ϵ-neighborhood of a point. Choosing the value of K is difficult due to presence of nonlinearity in the data and its variability in high dimensions. ELPP, incorporates K-Means for determining two points as neighbors, for the adjacency matrix construction. This helps in exploiting the natural grouping of the dataset. K-means classifier is first applied to the input training dataset for natural classes formation. Points belonging to same class are considered as neighbors by connecting them by an edge which is weighted by the new weighing function.

New weighing function used is the combination of two distribution functions: Heat Kernel and the Cauchy, for giving more emphasis to the overlapping regions. This gives more weight to moderately separated data points but it is not monotonically decreasing function, so Z-shaped function is used, which is undermentioned:

$$S_{i,j} = \begin{cases} 1; & \text{if } x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2; & \text{if } a \leq x \leq \frac{a+b}{2} \\ 2\left(b - \frac{x}{b-a}\right)^2; & \text{if } \frac{a+b}{2} \leq x \leq b \\ 0; & \text{otherwise} \end{cases}$$

Here a and b specify the range of the values along which the function changes its values and can be controlled and x is the distance between two data points. The final transformation matrix construction is done in the same way as in LPP.

2.3 Supervised Locality Preserving Projection (SLPP)

The objective of LPP is to preserve the intrinsic local geometry or the neighborhood relationship among the datapoints after their projection in lower dimen-

sional space. It projects the points of same class close to each other due to their closeness in original space. ELPP tries to overcome the ambiguity of overlapping region of classes as well. The reducibility is of use if the data is well classifiable after the projection or it can be clustered in the embedded space. With this new key, Supervised Locality Preserving Projection (SLPP) was proposed, which uses the class labels of data points to enhance the discriminating power besides inheriting the properties of LPP.

If data classes are close to each other or they have some overlapping region then neighboring points may belong to different classes. As per objective function of LPP, data points which are neighbors in original space will remain neighbor in reduced space also, even though they belong to different classes. This may lead to wrong classification of the data. The class information can be utilized to enhance the discriminant analysis as proposed in LDA. Here LPP can be modified with combination of class information with locality preserving property to enhance the classification. SLPP determines the neighborhood by using available class membership information so that neighbors belonging to different classes may be projected far enough for discrimination of the classes in reduced space as well.

SLPP is a supervised version of LPP that uses prior information of class labels of the data points. Unlike LPP, it determines the neighbors of a data point based on known class labels. Points belonging to same class are considered as neighbors and are projected close to each other in the embedded space. This helps in clustering the classes in reduced dimensions. SLPP uses the same objective function as of LPP but weight matrix is computed in a different way. LPP constructs the similarity matrix based on the nearest neighbors whereas SLPP uses the class information and makes points neighbors if they belong to the same class.

To make the computations simpler, the data samples are arranged compactly i.e. all the samples from the same class are arranged together in X which simplifies the computation of S. This makes X an orderly matrix , comprising of sub matrices A_i of size $n \times M$, $i = 1, 2, \ldots, c$, where c is the number of classes. Since SLPP considers points of same class as neighbors, so for each $x_j \in A_i$ are sought in A_i only. For each class, matrix B_i is calculated which will then be used for constructing S. Here, the weight is set to be 1 if two data points belong to the same

class otherwise weight is made. Hence, B_i turns out to be:

$$B_i = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix}$$

S is constructed by arranging the $B_i s$ for all the classes diagonally i.e.

$$S = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_c \end{bmatrix}$$

All other computations for finding the transformation matrix are same as done in LPP i.e. the transformation matrix w is found by solving the generalized eigenvalue problem : $XLX^T w = XDX^T w$. In SLPP, the distance between the points from same class is not taken into consideration. Hence, all the points are treated in the same way. In addition to the class information, if the weights assigned in S are based on the distance, it may be more helpful in revealing the underlying manifold.

2.4 Extended Supervised Locality Preserving Projection (ESLPP)

In SLPP, data points from same class are considered as neighbors and an equal weight 1 is assigned to all neighbors while constructing similarity matrix. SLPP does not inherit any property of the distance among the neighbors. Adding the information of the distance between the neighbors may help in enhancing the locality preservation capability.

Taking this additional information into account, an extended version of SLPP is formulated, known as ESLPP, where the weight matrix S is computed as per the proposal of ELPP. As in SLPP,while forming similarity matrix, two points are regarded as neighbors if they belong to the same class, instead of assigning same weights here the weight assigned is computed by a Z-shaped weighing function as shown in equation 2.2. This function assigns higher weight to the points belongong to the same class which are close enough in the original space and decreases weight as the distance increases.This helps in maintaining the internal structure

of the class in the projected space as well. All other computations are carried out the same way as that of SLPP.

$$S_{ij} = \begin{cases} \text{z-shaped weighing function;} & \text{if } x_i \text{ and } x_j \text{ belong to the same class} \\ 0; & \text{otherwise} \end{cases} \tag{2.2}$$

CHAPTER 3
Classifiers Explored

In Pattern Recognition, classification is the phenomena of labeling the data point (or a pattern) to a class from a set of possible classes. Classification may be of two types *Supervised* and *Unsupervised*. In the former type available information of the memberships of the training data points is used to classify a testing data point whereas, in the later one data points are classified by recognizing the underlying natural clustering among the available data points. The algorithm that implements the task of classification is called a **Classifier**. The classifiers which classify data using statistical inferences extracted from the training data are called **Statistical Classifiers**. These classifiers assign the probability to a data point being belonging to a possible class. Then the input data point is mapped to a class having highest probability of the membership. The problem can be formally stated as: Given a pattern X of dimension d having feature values $\{x_1, x_2, \ldots, x_d\}$ and a set of n classes $C = \{c_1, c_2, \ldots, c_n\}$, assign one of the classes c_i to X . The features are assumed to follow a probability density function (may be marginal or joint, depending on their mutual conditional dependence) conditioned with the class. Thus, a pattern is assumed to be an element drawn randomly from a probability distribution $p(X|c_i)$. If a decision boundary in the form of a hyperplane of d dimensional space can be found between different classes, and subsequently the given pattern can be classified by locating the region of class to which it belongs. This classification is done by formulating an objective function of the classifier also known as *discriminant function*.

Learning about different distributions of the data conditioned with classes by classifier is called *learning* or *training* of the classifier and the data used is called *training data*. In the following sections, a brief specification of two classifiers: Naive Bayes Classifier (NBC) and Support Vector Machine (SVM) is given. NBC works with two cases, assuming that data distribution is known and not known as well.

3.1 Naive Bayes Classifier

Assuming that distribution of the training data is available or known, the best known classifier that minimizes the probability of classification error is **Naive Bayes Clasiifier** [39]. Naive Bayes Classifier assumes that the features are *conditionally independent* of each other. In other words, the presence or absence of a particular feature is unaffected by other features. Moreover, the distribution of every feature given a class is independent of distribution of other features. This makes the classification task substantially simpler and it is empirically seen that the assumption of independence does not reduce significant accuracy of the classification of the real world data.

As discussed above, a classifier incorporates a discrimant function for classification across decision boundary. There are several well-known decision rules like Bayes Decision Rule, Maximum Likelihood rule and Neyman-Pearson rule for defining desicion boundary. Bayes decision rule for minimizing the misclassification cost can be stated as: assign input pattern to class c_i for which the conditional risk

$$R(c_i|x) = \sum_{j=1}^{n} L(c_i, c_j) * P(c_j|x) \tag{3.1}$$

is minimum, where $L(c_i, c_j)$ is the cost of misclassifying the pattern to class c_i when the true class is c_j and $P(c_j|x)$ is Posterior probability. If the loss function is $0/1$, the conditional risk becomes equivalent to conditional probability of the misclassification.

$$L(c_i, c_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \tag{3.2}$$

So, the Bayes decision rule can be simplified as:

$$P(c_i|x) > P(c_j|x) \text{ for all } j \neq i \tag{3.3}$$

The probability model adopted by a classifier is a conditional model: $P(C = c_i|F_1 = x_1, F_2 = x_2, \ldots, F_d = x_d)$ for a class variable C to which a random pattern X may belong conditioned with random variables of features f_1 through F_d. To compute the such an expression of probability, Bayes rule is used:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

which gives us:

$$P(C|F_1, F_2, \ldots, F_d) = \frac{P(F_1, F_2, \ldots, F_d|C) * P(C)}{P(F_1, F_2, \ldots, F_d)} \tag{3.4}$$

Theoretically, this can be seen as:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

In equation 3.4, the dinominator value remains same for every class given a particular pattern, while the numerator is the decisive quantity while comparing the probabilities of all classes. By using chain rule, the joint probability can be written as:

$$\begin{aligned}
P(C, F_1, \ldots, F_d) &\propto P(C).P(F_1, \ldots, F_d|C) \\
&\propto P(C)P(F_1|C)P(F_2, \ldots, F_d|C, F_1) \\
&\propto P(C)P(F_1|C)P(F_2|C, F_1)P(F_3, \ldots, F_d|C, F_1, F_2) \\
&\propto P(C)P(F_1|C)P(F_2|C, F_1)P(F_3|C, F_1, F_2) \ldots P(F_d|C, F_1, \ldots, F_{d-1})
\end{aligned}$$

It works well when number of features d is small but, when it is large the problem arises to calculate the conditional probabilities. The assumption of mutual conditional independence among the features simplifies the tedious task of computing the probability as:

$$\begin{aligned}
P(C|F_1, \ldots, F_d) &\propto P(C)P(F_1|C)P(F_2|C) \ldots P(F_d|C) \\
&\propto P(C) \prod_{i=1}^{d} P(F_i|C)
\end{aligned}$$

So, under independence assumption, conditional distribution over the class variable C can be expressed as under:

$$P(C|F_1, F_2, \ldots, F_d) = \frac{P(C) \prod_{i=1}^{d} P(F_i|C)}{P(F_1, \ldots, F_d)} \tag{3.5}$$

The accuracy of classification done by NBC is dependent on basically how well separated are the distributions of the training data and how well known they are. If the distribution of the data follows any standard distribution or assumed to be following, in the training phase the parameters are learned from them which are

futher used in the discriminant function of the classifier. This kind of classifier is called *Parametric Classifier*. If the distribution of the data is unknown then it is estimated by using a density estimation technique and then classification is done, which is known as *Nonparametric classifier*. Both approaches are expalined in following sections.

3.1.1 Parametric classification

As specified above, for classification, the corresponding distribution of the features conditioned with every class should be known. If, the data is unimodel i.e. the distribution has only one local maxima, the parameters of the distribution are used to determine the probability of any point randomly drawn from that distribution. If, underlying distribution of the features is assumed to be Gaussian, the parameters would be their respective means and variances. So, the process of classification is formulated as:

Training Given a training data set $Y = \{Y^1, Y^2, \ldots, Y^k\}$ each vector of dimension d, i.e. $Y^i = y^i_1, y^i_2, \ldots, y^i_d$ with their corresponding class labels. Let number of classes be m. Take data points labeled with every class and compute their mean vectors and variance vectors, say $\mu^1, \mu^2, \ldots, \mu^m$ and $\sigma^1, \sigma^2, \ldots, \sigma^m$ respectively.

Testing Once the training is done, the classifier is then used for classifying the test pattern. Using the parameters obtained for every class of data from the training phase, the problem reduces to:

$$\text{Class} = \arg\max_C P(C) \prod_{i=1}^{d} P(F_i|C)$$

where $P(F_i|C)$ is calculated based on the pdf of the data of every class. The class having maximum posteriori is assigned to the test pattern.

But, sometimes the data does not follow any standard distribution due to which there is no way of making use of any parameters. In such case, the phenomenon called non-parametric classification is used.

3.1.2 Non-parametric classification

Nonparametric classification is useful when the available training data does not follow any standard distribution or whose density function is not known. This may also happen when the data distribution follows a mixture model. In such

case, probability density function of the data has to be estimated with available data. One of the estimation technique is *Kernel Density Estimation* (KDE) or the *parzen window*.

Nonparametric estimation is a kind of histogram approximation technique. Let a set of data points is given, the histogram of the data points is formed. The x-axis is first divided into successive bins of width b. Probability of sample x being located in a bin is estimated for each of the bins. If total number of samples is N, samples located in a bin is k_N, the corresponding probability is approximated by the frequency ratio $P \approx k_N/N$. This approximation converges to the true density function P as $N \to \infty$. The corresponding pdf is assumed to be constant throughout the bin and is approximated by:

$$\hat{P}(x) \equiv \hat{P}(\hat{x}) \approx \frac{1}{h}\frac{k_N}{N}, |x - \hat{x}| \leq \frac{b}{2}$$

where b is the bin-width. This approach causes drastic changes in probability across bin boundaries as shown in figure 3.1.2.

To overcome this problem a smoothening algorithm is used known as *Kernel*

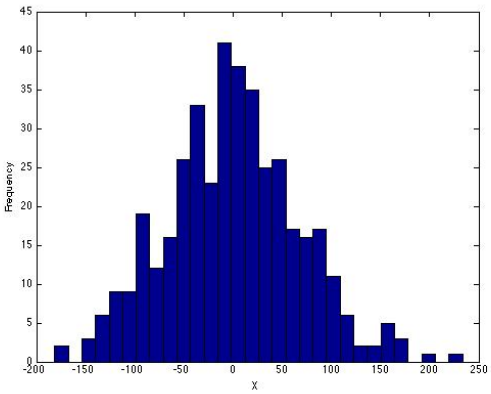

Figure 3.1: Histogram

Density Estimation, which is discussed in the following section.

Kernel Density Estimation (KDE)

KDE [40] is a fundamental data smoothening technique where inferences about the population are made, based on a finite data sample.
Method:

- Let $(x_1, x_2, \ldots, x_n)$ be n independently distributed samples drawn from some distribution with an unknown density f.

- We are interested in estimating the shape of this function f.
- Its kernel density estimator is:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.6}$$

  where $K(\cdot)$ is the Kernel and $h > 0$ is a smoothing parameter called the bandwidth. Different Kernel functions used are: uniform, triangular, triweight, normal and others.

The formal algorithm to estimate the density curve is as follows:

1. Let the given set of samples be $P = \{p_1, p_2, p_3, \ldots, p_n\}$.
2. Compute the range of this normalized data, i.e. $[Min(P), Max(P)]$.
3. Take m points $X = x_1, x_2, \ldots, x_m$ at equal interval within the computed range.
4. Calculate bandwidth $h = 1.06(\sigma_P)|P|^{-\frac{1}{5}}$, where σ_P is standard deviation of all points in P
5. For every point x_i, compute the probability density by summing up the densities calculated by the assumed (Gaussian) kernel with samples $x_j's$ as its mean and taking bandwidth h as standard deviation in equation 3.6, i.e.

$$p(x_i) = \frac{1}{nh}\sum_{j=1}^{n} e^{-\frac{1}{2}\left(\frac{x_i - p_j}{h}t\right)^2}$$

Plotting f against X, gives the required pdf.

Issues with KDE

- It is found that smoothness of the estimated curve is highly dependent on the bandwidth h used for the estimation. If the value of h is large then estimated curve is oversmooth and if h is small then the curve is found to be undersmooth.
- The estimated curve with bandwidth h [41] computed as per above expression may not be the best estimation for the classification purpose.

Tweak in Objective function of KDE

Experimentally, it was found that different values of bandwidth h gave different classification accuracy. Changing the constants in the calculation of h influenced the results. We varied two constants c_1 and c_2 in $h = c_1(\sigma_{V_b})|V_b|^{-\frac{1}{c_2}}$ within some specific ranges.

3.2 New Approach

Previous KDE method has bandwidth h, which is calculated using all the available data points. This makes the density estimation at a point affected by the whole distribution of the data points. Moreover, the value of h remains same for every point irrelevant of the total number of available data points present in the neighbor of the point. In other words, the presence of a cluster of data points can increase the density at a point near it despite of the observed density at that point is nearly zero. The new method is as below:
Let the available data points be $P = p_1, p_2, \ldots, p_n$ and the point at which density to be found be x.
Algorithm:

1. Choose optimal number of bins $'nb'$ as follows:
   - Take a large enough value $'nb\_max'$.
   - Check, if any bin in the histogram of P is empty, then decrease value of $'nb'$ and again repeat this step.
2. For every bin b of the histogram of P repeat following steps:
   - Form a vector V_b of all the points falling in the bin b.
   - If $|V_b|$ is less than 10 percent of $|P|$, then keep including points of left and right bins of b until the condition is met.
   - Let the observed frequency of bin b be $O_b = \textit{number of points in } b$.
   - Calculate the estimated frequency by

$$E_b = \left[\frac{1}{h}\sum_i e - \left(\frac{m_b - y_i}{h}\right)^2\right] * O_b \tag{3.7}$$

   where m_b is the mid-point of the bin b, y_i are the points in V_b and

$$h = c_1(\sigma_{V_b})|V_b|^{-\frac{1}{c_2}} \tag{3.8}$$

- Calculate $E_b(c_1, c_2)$ for different values for c_1 and c_2 and form a matrix of *goodness of fit* as:

$$gof(b, c_1, c_2) = \left(\frac{O_b - E_b(c_1, c_2)}{O_b} \right)^2 \tag{3.9}$$

3. Obtain *Goodness of fit* values corresponding to pair (c_1, c_2)

$$\chi = \sum_b gof(b, c_1, c_2) \tag{3.10}$$

4. Pick the values of c_1 and c_2 for which $\chi(c_1, c_2)$ is minimum, say $C1$ and $C2$.

5. Estimate the density at point x by

$$E_x = \frac{1}{h} \sum_i e^{-\left(\frac{x-y_i}{h}\right)^2} \tag{3.11}$$

b is the bin in which x falls, y_i are the points in V_b, and h is calculated as:

$$h = C1(\sigma_{V_b})|V_b|^{-\frac{1}{C2}} \tag{3.12}$$

3.3 Support Vector Machine

SVM is believed to be one of the best state-of-the-art supervised binary classifiers. SVM [42] works on the paradigm of finding the decision boundary based on the presence of largest margin between two classes and then classifying the data with respect to the boundary. Following subsections give the underlying fundamental for linearly separable and nonlinearly separable classes.

3.3.1 Linearly separable classes

Let $X = \{x^i : i = 1, 2, \ldots, n\}$ be a given set of training vectors. Note, that SVM works for classifying data belonging to only two classes. So, considering two classes to which a data point may belong to be ω^1 and ω^2. The problem is now to find a hyperplane $g(x) = w^T x + w_0$ that separates all the training vectors correctly as shown in the figure 3.3.1.

So, now the problem becomes of choosing the right hyperplane amongst the plausible hyperplanes. To resolve this problem, the optimum hyperplane has to be defined. The plane which satisfies the property of generalization performance

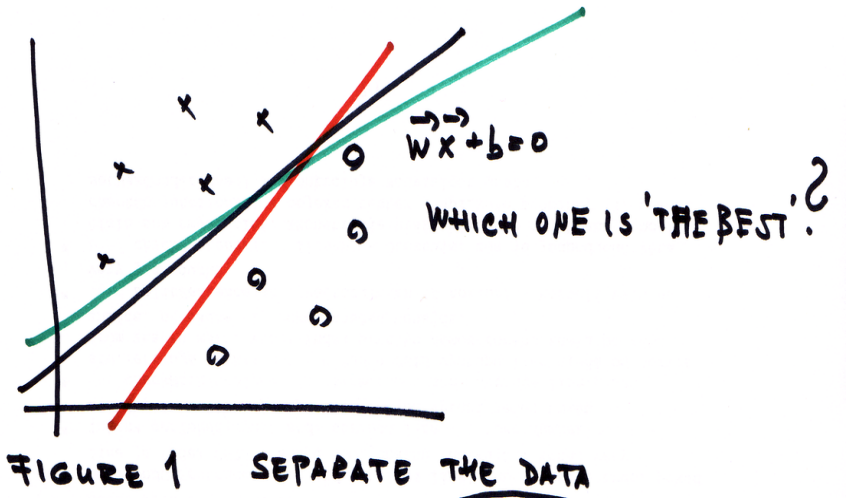

Figure 3.2: Possible classifying hyperplanes for a training data set. [1]

of the classifier. Its the capability of the classifier to classify a newly seen data point correctly based on the available information of the training data set. In other words, the hyperplane that leaves more "space" on either side, so that the data in both classes can move a bit more freely, with less risk of causing an error, should be chosen as shown in figure 3.3.1. A hyperplane is specified by its direction

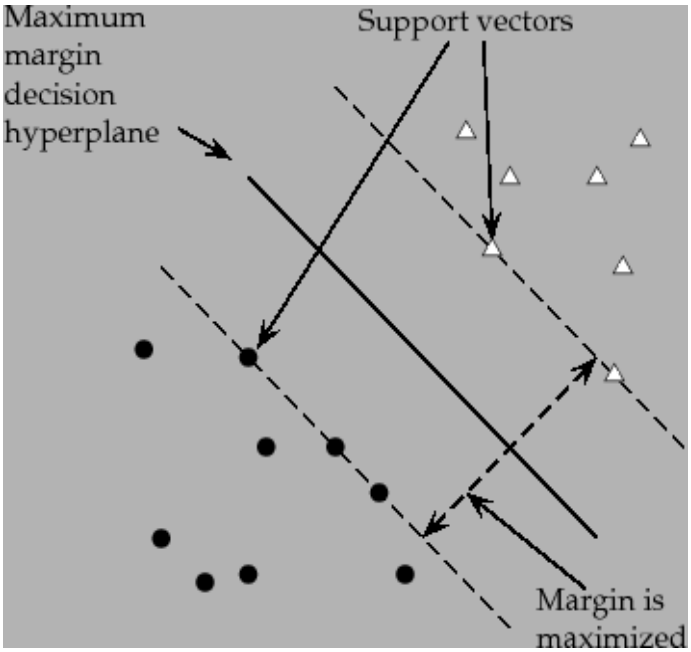

Figure 3.3: Optimum hyperplane. [2]

and w^T and its position in the space w_0. The direction along which the margin is maximum should be searched. Distance of the hyperplane from a point is given by:

$$z = \frac{|g(x)|}{||w||}$$

Now w and w_0 can be scaled so that the distance of $g(x)$ from the nearest point in the classes ω^1 and ω^2 is 1 and -1 respectively. Hence, the margin becomes $\frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||}$ such that,

$$w^T + w_0 \geq 1, \quad \forall x \in \omega^1 \tag{3.13}$$

$$w^T + w_0 \leq -1, \quad \forall x \in \omega^2 \tag{3.14}$$

As mentioned, training set comprises of labeled dataset, let y^i be the class label corresponding to the data point x^i, which is apparently either 1 for ω^1 or -1 for ω^2. This makes the above constraint to be more general as in equation 3.15 with the task which is:

$$\begin{aligned} &\text{Minimize} \quad J(w, w_0) \equiv \tfrac{1}{2}||w||^2 \\ &\text{Subject to} \quad y^i(w^T x^i + w_0) \geq 1, \quad i = 1, 2, \ldots, N \end{aligned} \tag{3.15}$$

Minimizing this norm makes the margin maximum.The Karush-Kuhn-Tucker (KKT) conditions that the minimizer has to satisfy are:

$$\frac{\partial}{\partial w} L(w, w_0, \lambda) = 0 \tag{3.16}$$

$$\frac{\partial}{\partial w_0} L(w, w_0, \lambda) = 0 \tag{3.17}$$

$$\lambda_i \geq 0, \quad i = 1, 2, \ldots, N \tag{3.18}$$

$$\lambda_i[y^i(w^T x_i + w_0) - 1] = 0, \quad i = 1, 2, \ldots, N \tag{3.19}$$

where λ is the vector of the Lagrange multipliers, λ_I, and $L(w, w_0, \lambda)$ is the Lagrangian function defined as:

$$L(w, w_0, \lambda) = \frac{1}{2} w_T w - \sum_{i=1}^{N} \lambda_i [y^i(w^T x_i + w_0) - 1] \tag{3.20}$$

Combining above equations, we get

$$w = \sum_{i=1}^{N} \lambda_i y^i x^i \tag{3.21}$$

$$\sum_{i=1}^{N} y^i x^i = 0 \tag{3.22}$$

The Lagrange multipliers can be either zero or positive. Thus, the vector parameter w of the optimal solution is a linear combination of $N_s \leq N$ features vectors that are associated with $\lambda_I \neq 0$. That is,

$$w = \sum_{i=1}^{N_s} \lambda_i y^i x^i \tag{3.23}$$

These are known as support vectors and the optimum hyperplane classifier as a *support vector machine* (SVM). A non zero Lagrange multiplier corresponds to an active constraint. Hence, as the set of constraints suggests for $\lambda_i \neq 0$, *the support vectors lie on either of the two hyperplanes*, that is,

$$w^T + w_0 = \pm 1 \tag{3.24}$$

In other words, they are training vectors that are closest to the linear classifier, and they constitute the critical elements of the training set.

Since, the optimization task belongs to the convex programming family of problems, and the cost function is convex and the constraints are linear and define a convex set of feasible solutions. Considering the Lagrangian duality can solve such problems. The problem can be stated equivalently by its Wolfe dual representation form, that is,

$$\text{Maximize } L(w, w_0, \lambda) \tag{3.25}$$

$$\text{Subject to } w = \sum_{i=1}^{N} \lambda_i y^i x^i \tag{3.26}$$

$$\sum_{i=1}^{N} \lambda_i y^i = 0 \tag{3.27}$$

$$\lambda \geq 0 \tag{3.28}$$

Once the optimal Lagrange multipliers have been computed, by maximizing above equation, the optimal hyperplane is obtained and w_0 by complementary slackness

conditions.

3.3.2 Non-separable Classes

When classes are non-separable, any attempt to draw a hyperplane will never end up with a class separation band with no data points inside it. The trainee features now belong to one of the following three categories:

1. Vectors that fall outside the band and are correctly classified.
2. Vectors falling inside the band and correctly classified and satisfy the inequality $0 \leq y^i(w^T x + w_0) < 1$.
3. Vectors that are misclassified and obey inequality $y^i(w^T x + w_0) < 0$.

All three cases can be treated under a single type of constraints by introducing a new set of variables, namely,

$$y_i[w^T x + w_0] \geq 1 - \xi_i \tag{3.29}$$

The first category of data corresponds to $\xi_I = 0$, the second to $0 < \xi_I \leq 1$, and the third to $\xi_I > 1$. The variables ξ_I are known as slack variables. The goal is now to make the margin as large as possible but at the same time to keep the number of points with $\xi > 0$ as small as possible. This is same as minimizing the cost function

$$J(w, w_0, \xi) = \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i \tag{3.30}$$

$$\text{Subject to} \qquad y^i[w^T x + w_0] \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, N \tag{3.31}$$

$$\text{and} \qquad \xi_i \geq 0, \qquad i = 1, 2, \ldots, N \tag{3.32}$$

where ξ is the vector of the parameters ξ_i and the parameter C is a positive constant that controls the relative influence of the two competing terms.

The problem is again a convex programming one, and the corresponding Lagrangian is given by

$$L(w, w_0, \xi, \lambda, \mu) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \mu_i \xi_i - \sum_{i=1}^{N} \lambda_i [y^i(w^T x + w_0) - 1 + \xi_i] \tag{3.33}$$

The corresponding KKT conditions are

$$\frac{\partial L}{\partial w} = 0 \quad \text{or} \quad w = \sum_{i=1}^{N} \lambda_i y^i x^i \tag{3.34}$$

$$\frac{\partial L}{\partial w_0} = 0 \quad \text{or} \quad \sum_{i=1}^{N} \lambda_i y^i = 0 \tag{3.35}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \text{or} \quad C - \mu_i - \lambda_i = 0, \ i = 1, 2, \ldots, N \tag{3.36}$$

$$\lambda_i [y^i (w^T x + w_0) - 1 + \xi_i] = 0, \ i = 1, 2, \ldots, N \tag{3.37}$$

$$\mu_i \xi_i = 0, \ i = 1, 2, \ldots, N \tag{3.38}$$

$$\mu_i \geq 0, \ \ \lambda_i \geq 0, \ i = 1, 2, \ldots, N \tag{3.39}$$

The associated Wolfe dual representation now becomes

$$\text{Minimize} \quad L(w, w_0, \lambda.\xi, \mu) \tag{3.40}$$

$$\text{Subject to} \quad w = \sum_{i=1}^{N} \lambda_i y^i x^i \tag{3.41}$$

$$\sum_{i=1}^{N} \lambda_i y^i = 0 \tag{3.42}$$

$$C - \mu_i - \lambda_i = 0, \ i = 1, 2, \ldots, N \tag{3.43}$$

$$\mu_i \geq 0, \lambda_i \geq 0, \ i = 1, 2, \ldots, N \tag{3.44}$$

Substituting the above equality constraints into Lagrangian, we get

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y^i y^j x^{iT} x^j \right) \tag{3.45}$$

$$\text{Subject to} \ \ 0 \leq \lambda_i \leq C, \quad i = 1, 2, \ldots, N \tag{3.46}$$

$$\sum_{i=1}^{N} \lambda_i y^i = 0 \tag{3.47}$$

The Lagrange multipliers corresponding to the points residing either within the margin or on the wrong side of the classifier, that is, $\xi_I > 0$, are all equal to the maximum allowable value C. Indeed at the solution, for $\xi_I \neq 0$ the KKT conditions give $\mu_I = 0$ leading to $\lambda_I = C$. In other words, these points have the largest possible "share" in the final solution $\mathbf{w}$.

CHAPTER 4
Experiments and Results

4.1 Data Description

For this work, features are extracted by doing ELPP and ESLPP projections [43] and most significant coefficients are selected for classification purpose.

4.1.1 YALE Face Database

Figure 4.1: Sample images from YALE face database.

The database [44] was constructed by Yale Centre for Computational Vision and Control with 5760 images of 10 subjects with different illumination and pose variations. For this work a subset of this database is used [43] which consists of facial images of 10 persons and 36 facial images per person with variations. A

sample of some images of this database is shown in the figure 4.1. There are 24 images of a person taken for the training purpose and 12 images for the testing set. The projections are obtained after performing some preprocessing over the images like cropping and resizing them.

4.1.2 DAIICT Face Database

Figure 4.2: Sample images from DAIICT face database.

Another database that has been used is DAIICT video database, in which images were obtained from extracted frames from the videos taken of different persons with varying expressions. We used 404 images per person for training the classifier and 40 images as the testing set.
For expression recognition, 4 expressions: angry, neutral, smiling and open-mouth are recorded per person and 101 images per expression were used in training dataset and 10 images per expression were used for testing dataset.

4.2 Results

Results with different classifiers are reported in the tables 4.1, 4.2, 4.3, 4.4 and 4.5. First row specifies the number of features used and first column specifies the name of methods. We used *KNN* with 1,3 and 5 neighbors, *NBC* with Gaussian distribution assumption, KDE and new KDE, and *SVM* as classifiers. Each method has been evaluated on 5 different test cases.

For NBC with KDE we have varied the values of c_1 from 0.3 to 1.2 with an interval of 0.05 and c_2 from 2.2 to 4.5 with an interval of 0.2. We have picked up the maximum value of the result corresponding to different values of c_1 and c_2.
For new approach of KDE the range of c_1 and c_2 same as above.

Table 4.1 shows results on YALE face database with most significant ELPP coefficients taken as features for Face recognition. NBC with new approach is performing better than the conventional KDE method. However, SVM outperforms

with less dimensions with 100% accuracy. Note, YALE face database is not suitable for facial expression recognition.

Table 4.2 and 4.3 show the results of DAIICT database for face and facial expression recognition using ELPP coefficients, respectively. The new approach gives comparable results for face recogniton and better than the conventional approach for facial expression recognition. Similarly, Table 4.4 and 4.5 show the results of DAIICT database for face and facial expression recognition using ESLPP coefficients, respectively. The proposed approach is not comparable with conventional approach for face recognition but found to be comparable for expression recognition. The SVM is performing superior in all cases.

Table 4.1: Results (in %) of classification of Faces - YALE database with ELPP coefficients as features.

Methods	5	10	15	20	30	50	100	150	200
KNN, K=3	62.22	56.67	74.44	83.33	82.22	78.89	77.78	76.67	74.44
KNN, K=5	61.11	60	68.89	70	68.89	68.89	70	70	67.78
NBC-Gaussian	50	38.89	50	54.45	55.56	54.44	63.33	58.89	62.22
NBC-KDE	16.67	15.56	17.78	17.78	17.78	18.89	18.89	18.89	21.11
NBC-newKDE	42.22	46.67	46.67	50	54.44	54.44	57.78	56.67	55.55
SVM	92.22	92.22	94.44	95.56	96.67	98.89	100	100	100

Table 4.2: Results (in %) of classification of Faces - DAIICT database with ELPP coefficients as features.

Methods	Test cases	5	10	15	20	30	50	80	100
KNN, K=1	Test 1	96.82	99.55	100	100	100	100	100	100
	Test 2	96.82	99.55	100	100	100	100	100	100
	Test 3	97.73	99.77	100	100	100	100	100	100
	Test 4	97.27	99.77	100	100	100	100	100	100
	Test 5	94.77	99.32	100	100	100	100	100	100
KNN, K=3	Test 1	96.82	99.77	100	100	100	100	100	100
	Test 2	96.82	99.77	100	100	100	100	100	100
	Test 3	97.95	100	100	100	100	100	100	100
	Test 4	97.27	100	100	100	100	100	100	100
	Test 5	95.68	99.32	99.77	100	100	100	100	100
KNN, K=5	Test 1	97.05	99.77	100	100	100	100	100	100
	Test 2	97.05	99.77	100	100	100	100	100	100
	Test 3	97.50	99.77	100	100	100	100	100	100
	Test 4	96.82	99.77	100	100	100	100	100	100
	Test 5	95.23	99.32	99.54	99.77	99.77	99.77	99.77	99.77
NBC-Gaussian	Test 1	86.36	86.59	88.86	88.63	90.45	95.68	91.13	90
	Test 2	84.54	86.81	89.09	88.86	90.68	95.91	91.36	90.23
	Test 3	85.45	86.59	90	89.54	91.14	95.91	92.50	91.13
	Test 4	83.64	85.45	88.18	88.64	89.32	95.23	92.05	90.45
	Test 5	82.50	83.86	87.27	86.13	89.77	95.68	91.36	89.09
NBC-KDE	Test 1	87.50	96.14	98.40	100	100	100	100	100
	Test 2	88.18	96.14	98.40	99.10	100	100	100	100
	Test 3	88.41	96.59	98.41	98.64	99.77	100	100	100
	Test 4	86.36	95.91	97.95	98.18	99.55	100	100	100
	Test 5	86.59	95.45	97.50	97.73	98.86	99.55	99.55	99.55
NBC-newKDE	Test 1	84.55	93.41	93.64	95.00	94.55	95.91	94.32	94.31
	Test 2	84.55	93.41	93.64	95.00	94.55	95.91	94.32	94.31
	Test 3	87.50	94.09	93.86	95.91	95.23	96.91	95.46	95
	Test 4	84.09	93.41	94.09	95.23	94.77	96.36	95.23	95.23
	Test 5	82.95	92.05	94.55	95.45	94.09	94.77	92.73	92.27
SVM	Test 1	97.05	100	100	100	100	100	100	100
	Test 2	97.05	100	100	100	100	100	100	100
	Test 3	97.50	100	100	100	100	100	100	100
	Test 4	97.50	99.55	99.55	99.55	99.55	99.55	99.55	99.55
	Test 5	96.60	97.73	99.55	99.09	99.55	99.55	99.55	99.55

Table 4.3: Results (in %) of classification of Expressions - DAIICT database with ELPP coefficients as features.

Methods	Test cases	5	10	15	20	30	50	80	100
KNN, K=1	Test 1	68.18	72.50	72.95	73.41	73.41	72.05	72.05	71.59
	Test 2	70.91	74.09	74.77	74.32	73.64	74.09	74.55	74.55
	Test 3	74.77	78.64	79.55	78.86	78.41	79.09	79.09	79.32
	Test 4	75.45	79.77	79.55	79.77	79.77	78.86	79.55	79.09
	Test 5	72.73	77.95	78.41	78.18	78.41	77.73	78.18	77.73
KNN, K=3	Test 1	67.73	72.05	70.91	71.36	71.14	70.68	69.32	68.18
	Test 2	68.86	73.64	73.64	73.86	72.73	73.18	72.50	71.82
	Test 3	72.50	77.95	78.18	78.41	77.50	78.18	77.27	76.59
	Test 4	76.14	79.09	79.32	79.09	79.09	77.73	76.14	75.68
	Test 5	72.95	77.05	77.95	78.18	78.64	77.50	75	75
KNN, K=5	Test 1	67.27	70	69.09	69.55	68.18	67.05	66.82	66.14
	Test 2	69.09	72.95	72.27	71.59	71.14	69.32	69.55	69.10
	Test 3	73.18	77.27	77.05	76.14	75.91	74.09	74.32	73.86
	Test 4	74.55	77.50	78.18	77.73	77.27	75.91	75	74.32
	Test 5	72.50	76.59	77.50	77.05	77.50	75.91	75.23	74.10
NBC-Gaussian	Test 1	43.18	47.95	46.82	49.10	47.05	43.86	42.27	39.55
	Test 2	46.14	51.82	50	50	49.55	46.82	42.50	41.59
	Test 3	47.95	53.18	51.36	50.91	50.23	48.18	43.41	42.50
	Test 4	47.27	43.41	43.18	54.09	52.73	48.86	44.09	42.72
	Test 5	48.86	54.32	52.05	53.18	52.27	49.77	43.41	41.59
NBC-KDE	Test 1	42.73	50	50.67	51.84	52.34	48.88	46.92	46.12
	Test 2	43.41	50.23	50.80	53.68	54.07	49.80	50.61	48.30
	Test 3	45.23	52.27	52.27	54.09	53.86	50.45	48.41	47.05
	Test 4	47.73	56.14	56.24	55.58	54.87	52.45	50.38	49.31
	Test 5	48.64	55	55.20	56.04	55.92	53.93	51.72	50.84
NBC-newKDE	Test 1	39.77	50.45	49.55	52.27	50.68	70.38	69.30	70.91
	Test 2	42.05	52.27	51.59	51.59	50	70.68	68.18	67.27
	Test 3	44.32	54.32	53.41	53.41	51.82	72.27	69.77	68.64
	Test 4	46.59	54.09	54.32	54.55	53.64	71.14	70.23	69.09
	Test 5	47.05	54.09	54.09	55.23	53.41	72.73	70.91	69.77
SVM	Test 1	92.05	92.05	92.17	92.05	92.27	91.59	91.36	90
	Test 2	91.59	90.45	90.68	90	90	89.77	89.32	87.95
	Test 3	94.09	94.32	94.55	94.09	94.09	93.86	93.41	92.05
	Test 4	94.55	94.77	94.77	93.86	94.09	93.86	93.64	92.50
	Test 5	93.86	94.32	94.32	93.41	93.41	93.18	92.73	91.36

Table 4.4: Results (in %) of classification of Faces - DAIICT database with ESLPP coefficients as features.

Methods	Test cases	5	10	15	20	30	50	80	100
KNN, K = 1	Test 1	100	100	100	100	100	100	100	100
	Test 2	100	100	100	100	100	100	100	100
	Test 3	100	100	100	100	100	100	100	100
	Test 4	100	100	100	100	100	100	100	100
	Test 5	97.72	100	100	100	100	100	100	100
KNN, K = 3	Test 1	100	100	100	100	100	100	100	100
	Test 2	100	100	100	100	100	100	100	100
	Test 3	100	100	100	100	100	100	100	100
	Test 4	100	100	100	100	100	100	100	100
	Test 5	97.72	100	100	100	100	100	100	100
KNN, K = 5	Test 1	100	100	100	100	100	100	100	100
	Test 2	100	100	100	100	100	100	100	100
	Test 3	100	100	100	100	100	100	100	100
	Test 4	100	100	100	100	100	100	100	100
	Test 5	97.72	100	100	100	100	100	100	100
NBC-Gaussian	Test 1	92.5	98.86	100	100	100	100	100	100
	Test 2	92.5	98.86	100	100	100	100	100	100
	Test 3	93.86	98.40	100	100	100	100	100	100
	Test 4	93.86	98.40	100	100	100	100	100	100
	Test 5	91.13	98.86	100	100	100	100	100	100
NBC-KDE	Test 1	95.45	100	100	100	100	100	100	100
	Test 2	95.45	100	100	100	100	100	100	100
	Test 3	95	100	100	100	100	100	100	100
	Test 4	95	100	100	100	100	100	100	100
	Test 5	95	100	100	100	100	100	100	100
NBC-newKDE	Test 1	62.5	67.5	70.45	69.09	68.63	69.77	70.68	70.45
	Test 2	65.22	69.54	73.18	72.27	71.36	72.5	73.63	73.63
	Test 3	66.81	72.27	73.40	74.09	73.40	74.54	75.45	75.90
	Test 4	65.22	68.40	70.68	71.13	71.13	73.86	74.77	74.54
	Test 5	58.63	60.68	64.31	66.36	65.68	67.04	68.63	67.72
SVM	Test 1	100	100	100	100	100	100	100	100
	Test 2	100	100	100	100	100	100	100	100
	Test 3	100	100	100	100	100	100	100	100
	Test 4	100	100	100	100	100	100	100	100
	Test 5	97.72	100	100	100	100	100	100	100

Table 4.5: Results (in %) of classification of Expressions - DAIICT database with ESLPP coefficients as features.

Methods	Test cases	5	10	15	20	30	50	80	100
KNN, K = 1	Test 1	74.09	75.90	76.81	77.72	76.36	76.13	77.5	77.72
	Test 2	78.18	78.86	81.13	81.81	79.54	79.77	81.36	81.81
	Test 3	77.5	78.63	81.36	81.59	80	80.22	81.81	81.36
	Test 4	79.09	80.22	82.72	82.5	81.13	81.13	82.72	82.27
	Test 5	75	75.22	78.18	77.27	76.36	75.91	77.95	77.27
KNN, K = 3	Test 1	73.18	75.22	77.95	77.72	76.13	76.36	77.04	77.5
	Test 2	77.5	78.18	81.81	81.81	80	80.68	81.59	82.04
	Test 3	77.5	77.95	82.27	81.13	80.45	80	81.59	81.59
	Test 4	78.63	79.31	82.04	82.72	80.90	80	81.81	82.5
	Test 5	75.22	73.86	77.95	76.13	76.59	75.90	77.5	77.27
KNN, K = 5	Test 1	73.41	74.54	77.5	77.04	76.59	77.27	77.5	78.18
	Test 2	77.5	77.72	81.59	81.59	80.68	81.36	82.27	82.5
	Test 3	77.72	77.72	82.27	80.45	80.90	80	81.81	82.04
	Test 4	78.63	78.86	81.59	82.27	79.77	80.22	81.81	82.72
	Test 5	73.40	73.86	78.18	76.81	75.90	76.13	77.27	77.5
NBC-Gaussian	Test 1	61.81	66.81	67.72	68.63	68.18	68.18	71.36	71.818
	Test 2	64.54	70	70.68	71.59	71.36	71.36	74.31	74.31
	Test 3	67.04	71.36	72.04	72.04	72.95	73.18	76.59	75.90
	Test 4	65.22	70.90	71.81	71.81	72.27	72.27	75.90	75
	Test 5	59.54	64.09	65.45	65	65.68	66.59	69.77	67.95
NBC-KDE	Test 1	65.22	67.95	70	70	70.45	70.45	71.81	72.72
	Test 2	67.27	70.22	72.27	72.04	73.40	72.95	74.54	74.54
	Test 3	70.45	72.04	73.18	74.31	75.22	74.77	77.04	76.36
	Test 4	69.09	69.77	72.5	72.95	74.54	75	77.27	76.36
	Test 5	62.5	62.04	66.81	67.04	67.72	68.18	68.86	69.31
NBC-newKDE	Test 1	62.5	67.5	70.45	69.09	68.63	69.73	70.68	70.45
	Test 2	65.22	69.54	73.18	72.27	71.36	72.5	73.63	73.63
	Test 3	66.81	72.27	73.40	74.09	73.40	74.54	75.45	75.90
	Test 4	65.22	68.40	70.68	71.13	71.13	73.86	74.77	74.54
	Test 5	58.63	60.68	64.31	66.36	65.68	67.04	68.63	67.72
SVM	Test 1	83.86	84.77	84.54	82.04	79.54	82.95	84.09	82.95
	Test 2	87.27	88.18	88.40	87.27	84.77	88.18	89.31	87.95
	Test 3	87.72	88.40	88.18	87.5	85.22	88.86	90.22	90
	Test 4	88.63	90.22	89.31	87.04	85.45	88.86	89.31	89.09
	Test 5	84.31	85.45	86.36	83.18	80.22	83.86	85.22	84.54

CHAPTER 5
Conclusion and Future Work

5.1 Conclusion

The advancement of machine learning methods made possible to involve numerous challenges of face recognition system in camparison to few decades back. Unfortunately, still research in this area lack in 100% efficiency and in terms of robustness and also computation complexity as the number of users in the system increases. Most of the approaches restrict those constraints and find the suitable solution for given situation.

In this thesis, the emphasis has been given on dimensionality reduction based methods for face recognition and facial expression classification under restricted environment. The method derives the features for faces and facial expressions by using Locality Preserving Projection which is far superior projection methods in the literature. Using these features, a statistical classifier, Naive Bayes Classifier is run. The purpose is to classifiy each face image (data point) to its similar face or to the same person's faces available in the dataset. Similar, experiments have been performed for facial expression recognition task where one's facial expression can be classified simulantanously while projecting the face image.

The Face recognition and expression recognition problem have been addressed here from Density estimation point of view of available pattern recognition domain. Density Estimation of the features with unknown distribution is a vital task for classification based on statistical techniques. Kernel Density Estimation is one of the best known technique for the same. There is no standard rule for bandwidth selection claimed in the literature other than Silverman's thumb rule. Experimentally, it was found that change in parameters of bandwidth computation as per the rule gave different results of classification. A new approach is proposed for density estimation of the features for better classification results produced by us-

ing Naive Bayes classifier. This new approach takes only neighboring datapoints into account for computing the bandwidth in addition to the goodness of fit of the estimated curve.

The experiments have been conducted on YALE face database and DAIICT database to validate proposed methodology. The databases involve multiple persons including male and female persons with various facial expressions. Although, the major limitation of the work is the availability of large database and experiments on real time video processing. The contribution of this work is summerized as below:

- Results produced with new approach are far more better than the KDE on the YALE face database.
- SVM outperformed other classifiers used in this work, namely NBC and KNN on both the databases.
- Considering the mutual conditional independence assumption of NBC, it may not be the best classifier for the correlated data.

5.2 Future Work

The Face recognition problem is modeled as an statisitical estimation problem. The input of this estimation problem is the output of the coefficients of the ELPP and ESLPP methods having predefined number of dimensions. The ELPP and ESLPP methods has shown their robustness performance in the literature in comparison to their contemporary approaches. So, these projection methods implicitly provide a robust approach against outlier. However, one can use an explicit regularizer in the estimation procedure to add second level of robustness on the output of the projection space.

Although, a new approach has been proposed in this thesis for improving classical Navie Bayes Classifier but experiment only to face recognition and facial expression recognition problem. The proposed approach can extensively be applied in different area of Machine Learning problem.

The problem of face recogniton and expression recognition is more challenging in real time video implementation or surviellence based systems which has not been addressed in this work.

References

[1] B. V. HladkÃą, "Support vector machines," http://ufal.mff.cuni.cz/~hladka/ML/LECTURES/jsmath/test/support-vector-machines.html, Feb. 2007.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[3] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 4–37, 2000.

[4] M. D. Kelly, "Visual identification of people by computer." DTIC Document, Tech. Rep., 1970.

[5] T. Kanade, *Computer recognition of human faces*. Birkhäuser, 1977, vol. 47.

[6] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 23–38, 1998.

[7] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 39–51, 1998.

[8] L. Gu, S. Z. Li, and H.-J. Zhang, "Learning probabilistic distribution model for multi-view face detection," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–116.

[9] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 45–51.

[10] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?" *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245–260, 1998.

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

[12] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 6, pp. 748–763, 2002.

[13] M.-H. Yang and N. Ahuja, "Detecting human faces in color images," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1. IEEE, 1998, pp. 127–130.

[14] P. W. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 995–999.

[15] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham *et al.*, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[16] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.

[17] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic face identification system using flexible appearance models," *Image and vision computing*, vol. 13, no. 5, pp. 393–401, 1995.

[18] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 681–685, 2001.

[19] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 1, pp. 103–108, 1990.

[20] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 696–710, 1997.

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

[22] H. Moon and P. J. Phillips, "Computational and performance aspects of pca-based face-recognition algorithms," *Perception-London*, vol. 30, no. 3, pp. 303–322, 2001.

[23] D. L. Swets and J. J. Weng, "Using discriminant eigenfeatures for image retrieval," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 831–836, 1996.

[24] 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.

[25] W. Zhao, R. Chellappa, and P. J. Phillips, *Subspace linear discriminant analysis for face recognition*. Citeseer, 1999.

[26] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski, "Independent component representations for face recognition," 1998.

[27] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[28] ——, "The independent components of natural scenes are edge filters," *Vision research*, vol. 37, no. 23, p. 3327, 1997.

[29] S. Kung and J. Taur, "Decision-based neural networks with signal/image classification applications," *Neural Networks, IEEE Transactions on*, vol. 6, no. 1, pp. 170–181, 1995.

[30] 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.

[31] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg, "The bochum/usc face recognition system and how it fared in the feret phase iii test," in *Face Recognition*. Springer, 1998, pp. 186–205.

[32] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 775–779, 1997.

[33] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *Computers, IEEE Transactions on*, vol. 42, no. 3, pp. 300–311, 1993.

[34] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 84–91.

[35] D. L. Ruderman, "The statistics of natural images," *Network: computation in neural systems*, vol. 5, no. 4, pp. 517–548, 1994.

[36] J. Huang, B. Heisele, and V. Blanz, "Component-based face recognition with 3d morphable models," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 27–34.

[37] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, March 2003.

[38] G. Shikkenawis and S. Mitra, "A new proposal for locality preserving projection," in *PerMIn*, 2012, pp. 298–305.

[39] K. P. Murphy, "Naive bayes classifiers," 2006.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley-interscience, 2012.

[41] B. W. Silverman, *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1986, vol. 26.

[42] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier, 2009.

[43] G. Shikkenawis, *Locality Preserving Projection : A Study and Applications*. M.Tech. Dissertation, DAIICT, 2012.

[44] A. Georghiades, "Yale face database," *Center for Computational Vision and Control at Yale University.[Online]. Available: http://cvc. yale. edu/projects/yalefaces/yalefaces. html*, 1997.