Spring 2011

Learning Local Features Using Boosted Trees for Face Recognition

Rajkiran Gottumukkal
Old Dominion University

LEARNING LOCAL FEATURES USING BOOSTED TREES FOR FACE RECOGNITION

by

Rajkiran Gottumukkal
M.S. May 2003, Old Dominion University

A Dissertation submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2011

Approved by:

Vijayan K. Asari (Director)

Shirshak K. Dhali (Member)

Gene Hou (Member)

Jiang Li (Member)

ABSTRACT

LEARNING LOCAL FEATURES USING BOOSTED TREES FOR FACE RECOGNITION

Rajkiran Gottumukkal
Old Dominion University, 2011
Director: Dr. Vijayan K. Asari

Face recognition is fundamental to a number of significant applications that include but not limited to video surveillance and content based image retrieval. Some of the challenges which make this task difficult are variations in faces due to changes in pose, illumination and deformation. This dissertation proposes a face recognition system to overcome these difficulties. We propose methods for different stages of face recognition which will make the system more robust to these variations. We propose a novel method to perform skin segmentation which is fast and able to perform well under different illumination conditions. We also propose a method to transform face images from any given lighting condition to a reference lighting condition using color constancy. Finally we propose methods to extract local features and train classifiers using these features. We developed two algorithms using these local features, modular PCA (Principal Component Analysis) and boosted tree. We present experimental results which show local features improve recognition accuracy when compared to accuracy of methods which use global features.

The boosted tree algorithm recursively learns a tree of strong classifiers by splitting the training data in to smaller sets. We apply this method to learn features on the intra-personal and extra-personal feature space. Once trained each node of the boosted tree will

be a strong classifier. We used this method with Gabor features to perform experiments on benchmark face databases. Results clearly show that the proposed method has better face recognition and verification accuracy than the traditional AdaBoost strong classifier.

This dissertation is dedicated to my family and friends.

ACKNOWLEDGMENTS

I feel lucky to have this opportunity of pursuing a doctoral degree, and I am grateful to all those who made it possible.

I would like to thank my advisor, Dr. Vijayan Asari for his support and guidance throughout this endeavor. He was always ready with good advice and honest feedback. He was also my master's degree advisor and introduced me to the areas of image processing and computer vision. Without his patience and encouragement I may have never discovered the joy of research.

I am grateful to the faculty and staff of Electrical and Computer Engineering department at Old Dominion University for their support and funding provided over the years. Special thanks also to the members of my thesis committee, thank you for being flexible in accommodating me.

I would like to thank my parents for encouraging me to focus on education and supporting my decision to pursue my graduate degree. Last but not least, thanks to my wife Bindu. The work that has gone into this thesis has been tedious and painful, thanks for the support and help you gave me to get through this journey.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

I. INTRODUCTION

Human visual system is capable of very robust object recognition. Humans can recognize a familiar object even after its appearance has changed due to pose, lighting conditions, deformation and/or occlusion. Over the years several researchers have tried to replicate this capability in computers with varying degree of success. Automating this process has several important applications.[1]

One of the recent applications is, being able to recognize objects of interest in surveillance videos. Surveillance cameras are being deployed in large numbers all over the world to prevent crime from happening. Typically a human operator is needed to observe the video feeds to detect objects of interest. However this has the disadvantage of human error due to fatigue and shortage of man power. Computerized object recognition can alleviate these problems; however it has to be as robust as human visual processing. A robust object recognition system can aid human operator by drawing his/her attention to the objects of interest without too many false recognitions. Human operator can then use his/her cognition to determine the appropriate action to be taken.

Another important application is in automated manufacturing quality control. For example a bottling factory could be manufacturing several thousand bottles per hour. Some of the bottles may have manufacturing defects, for example the label on a bottle may be missing. If a human operator has to check each bottle for defects the manufacturing process will slow down. On the other hand a computer can analyze images

[1] This work is written as per the *IEEE Transactions* format

from a camera at a much faster rate. With a computer capable of robustly recognizing defects the manufacturing will not be slowed down by the quality control.

Content based image retrieval (CBIR) is a broad area where object recognition is used. The problem involves retrieving a set of closely matching images from a large database of images given a query image. In the absence of CBIR, humans have to manually search the database to find images similar to the query image. This is too tedious even for a database having few hundred images and impractical for databases having millions of images. CBIR has several applications such as law enforcement, medical images retrieval, and retail.

A. Problem Statement

Given an input image, find the images from a database which are visually most similar to the input image. To clarify this further, we consider the example of face recognition. Typically we have 2D face images of different individuals, which may be acquired under controlled conditions with arbitrary camera and environmental conditions. Also, each individual may have multiple instances of their face image. Now, given a test image the computer has to compute matching scores of the test image against images in the database. The highest matching score should be given to the image which is most similar to the test image.

Under constrained conditions satisfactory accuracy has been achieved to deploy the facial recognition system in real world applications. However, unconstrained variations in imaging conditions result in less than satisfactory accuracy. The major factors that influence the recognition process are listed below.

- Pose – Pose of the face can vary with respect to position of camera. Appearance of face varies a lot from frontal to profile pose. When the database has faces at a particular pose and test image is at a vastly different pose, recognition is difficult.
- Orientation – Rotation of a face with respect to optical axis of camera causes features of face like eyes, mouth etc., to appear at different location.
- Occlusion – When parts of face are hidden from view of camera occlusion occurs. Occlusion can be of varying degree and at extreme cases where most of the face is occluded it may be impossible to recognize it. Facial hair and eyewear can also be considered as special cases of occlusion.
- Illumination – Images represent the intensity level of light received by the camera. This can vary with changes in position of light source, number of light sources and type of light sources.
- Deformation – Structure of face can change due to changes in expression. Any non-rigid object for that matter will appear different due to deformation.

These factors affect all form of object recognition. For an object recognition system to be practical it must address these issues. The work presented in this dissertation will try to directly address them.

B. Object Recognition System

In this section we discuss the basic building blocks of a general object recognition system. To explain certain aspects of the system will use the example of face detection application.

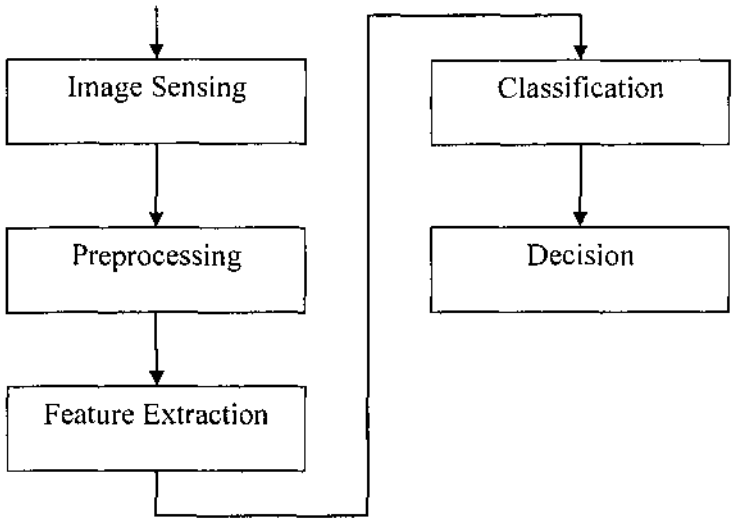


Fig. 1. General object recognition system.

1) Image Sensing

The real world scene is converted into digital data which can be used by a computer. This involves projecting the 3D real world points to 2D image plane. This is done using a digital or video camera. There is loss of information when we map the 3D points to the 2D image plane. The characteristics of camera also affect the quality of the captured image which can make the recognition task more difficult.

2) Preprocessing

Preprocessing is done to change the input images to a form more suitable for processing in the future stages. The methods used for preprocessing can vary based on the application and algorithms being used in the feature extraction and classification stages. One of the common preprocessing steps is to normalize the input images intensity using histogram equalization.

3) Feature Extraction

Feature extraction is done to transform the image data to a format which makes classification more accurate. Using the pixel intensities as features is not suitable for most classification tasks. This is due to high dimensionality of the image space and also pixel intensities at a give location can vary widely for images of the same class. In practice feature extrication and classification tasks are tightly coupled and they depend on the application.

4) Classification

The classifier is responsible for making the decision of which class the input image belongs. To make this decision, the classifier has to have prior knowledge of each class. This process is called training, training can be done either in supervised manner where we provide it with labeled examples or unsupervised manned where the classifier learns the labels from the underlying structure of the data.

C. Approaches

There are two main approaches to face recognition. They are model-based recognition and appearance-based recognition. In model-based approaches, a model of the face is created which can undergo geometrical transformation to map the model into sensors coordinate system. If a close match is found between the model and original image, the image is classified as belonging to the same class as the model. Appearance based approaches aim to learn the appearance of the face using training examples. In practice, salient features of the face are used as input to machine learning algorithms to learn a classifier which can distinguish between the trained face against other faces.

One of the advantages of model-based approaches is that a good match can be obtained from a few features due to strong geometric constraints. Due to this they are robust to factors like pose variations and occlusions. The downside of these methods is that it is time consuming to obtain the models and they are not general. On the other hand appearance-based methods are more general and can be easily tuned by adjusting training data. However they have the disadvantage of being sensitive to factors like changing illumination, expression, pose and occlusion which make it difficult to find matching features. My research comes under appearance-based approach, but models local appearance of parts and the relationships among the parts. We aim to make it robust to the presence of mentioned factors.

D. Thesis Outline

In this chapter, we provided the motivation for computer based face recognition and discussed some of the challenges that need to be overcome. In chapter 2, we will discuss in more detail the feature extraction and classification algorithms used in appearance-based methods. In chapter 3, we will discuss in detail the methods we developed for pose, illumination and expression invariant face recognition. In chapter 4, we will discuss in detail the boosted tree algorithm. Experimental results are shown in chapter 5, and in chapter 6 we will present our conclusions and ideas for future improvements.

E. Objectives

The main objectives of this research work are to make face recognition robust to changes in pose, orientation, occlusion, illumination, and expression. To achieve this we made improvements to the different building blocks of the face recognition system. We

have proposed a new method to perform skin segmentation which is fast and able to perform well under different illumination conditions. We have also proposed a method to transform face images from give lighting condition to a reference lighting condition using color constancy. Finally, we propose two approaches to face recognition using local features. We tested all the proposed methods thoroughly with benchmark datasets and analyzed the results.

II. APPEARANCE-BASED FACE RECOGNITION

There are two main categories of methods to do appearance-based face recognition, generative methods and discriminant methods. Generative methods such as principal component analysis (PCA) [1], independent component analysis (ICA) [2] or non-negative matrix factorization (NMF) [3] try to find a suitable representation of the original data. In contrast, discriminant methods such as linear discriminant analysis (LDA) [4], support vector machines (SVM) [5], or boosting [6] where designed for classification tasks. Given the training data and the corresponding labels the goal is to find optimal decision boundaries. Thus, to classify an unknown sample using a discriminative model a label is assigned directly based on the estimated decision boundary. In contrast, for a generative model the likelihood of the sample is estimated and the sample is assigned the most likely class.

In appearance-based methods images are usually captured by different two-dimensional views of the face of interest. Based on the applied features these methods can be sub-divided into two main classes, i.e., local and global approaches. A local feature is a property of an image located on a single point or small region. It is a single piece of information describing a rather simple, but ideally distinctive property of the face. Examples for local features of a face are the color, gradient or gray value of a pixel or small region. For face recognition tasks the local feature should be invariant to illumination changes, noise, scale changes and changes in viewing direction, but in general this cannot be reached due to the simplicity of the features itself. Thus, several features of a single point or distinguished region in various forms are combined and a

more complex description of the image usually referred to as descriptor is obtained. A distinguished region is a connected part of an image showing a significant and interesting image property. It is usually determined by the application of a region of interest detector to the image. In contrast, global features try to cover the information content of the whole image or patch. This varies from simple statistical measures (e.g., mean values or histograms of features) to more sophisticated dimensionality reduction techniques, i.e., subspace methods, such as principle component analysis (PCA) [1], independent component analysis (ICA) [2], or non negative matrix factorization (NMF) [3]. The main idea of all of these methods is to project the original data onto a subspace that represents the data optimally according to a predefined criterion.

Since the whole data is represented, global methods allow reconstructing the original image thus providing robustness to some extent. Contrary to this, due to the local representation, local methods can cope with partly occluded faces considerably better.

A. Feature Extraction

1) Local Features

These features describe the region or its local neighborhood by certain invariance properties. An important problem is the high dimensionality of these features for matching and recognition tasks. The computational effort is very high and thus, it is very important to reduce the dimensionality of the descriptor by keeping their discriminative power. We give a brief overview about some of the important state of the art methods.

- SIFT Descriptor

One of the most popular descriptors is the one developed by David Lowe [7, 8]. Lowe developed a carefully designed combination of detector and descriptor with excellent performance as shown in [9]. The detector/descriptor combination is called scale invariant feature transform (SIFT) and consists of a scale invariant region detector called difference of Gaussian (DoG) detector and a proper descriptor often referred to as SIFT-key.

The DoG-point detector determines highly repetitive interest points at an estimated scale. To get a rotation invariant descriptor, the main orientation of the region is obtained by a 36 bin orientation histogram of gradient orientations within a Gaussian weighted circular window. Note, that the particular gradient magnitudes m and local orientations Φ for each pixel $I(x, y)$ in the image are calculated by simple pixel differences according to

$$m = \sqrt{\big(I(x+1,y) - I(x-1,y)\big)^2 + \big(I(x,y+1) + I(x,y-1)\big)^2}$$

$$\emptyset = \tan^{-1}\big(\big(I(x,y+1) + I(x,y-1)\big)/\big(I(x+1,y) - I(x-1,y)\big)\big) \qquad (2.1)$$

The size of the respective window is well defined by the scale estimated from the DoG point detector. It is possible, that there is more than one main orientation present within the circular window. In this case, several descriptors on the same spatial location are created. For the descriptor all the weighted gradients are normalized to the main orientation of the circular region. The circular region around the key-point is divided into 4 x 4 non overlapping patches and the histogram gradient orientations within these patches are calculated. Histogram smoothing is done in order to avoid sudden changes of

orientation and the bin size is reduced to 8 bins in order to limit the descriptor's size. This results into a 4 x 4 x 8 = 128 dimensional feature vector for each key-point. Fig. 2 illustrates this procedure for a 2 x 2 window. Finally, the feature vector is normalized to unit length and thresholded in order to reduce the effects of linear and non-linear illumination changes.

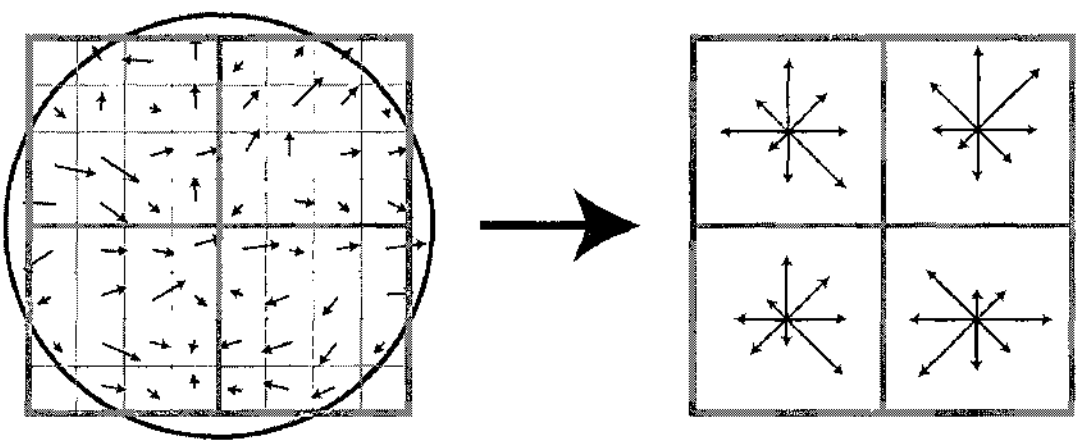


Fig. 2. Illustration of SIFT descriptor calculation, from these 2 x 2 patches a 32 dimensional histogram is calculated. [8]

- PCA-SIFT

Ke and Sukthankar [10] modified the DoG/SIFT-key approach by reducing the dimensionality of the descriptor. Instead of gradient histograms on DoG-points, the authors applied Principal Component Analysis (PCA) to the scale-normalized gradient patches obtained by the DoG detector. In principle they follow Lowe's approach for key-point detection. They extract a 41 x 41 patch at the given scale centered on a key-point, but instead of a histogram they describe the patch of local gradient orientations with a PCA representation of the most significant eigenvectors (that is, the eigenvectors

corresponding to the highest eigenvalues). In practice, it was shown, that the first 20 eigenvectors are sufficient for a proper representation of the patch. The necessary eigenspace can be computed off-line (e.g., Ke and Sukthankar used a collection of 21.000 images). In contrast to SIFT-keys, the dimensionality of the descriptor can be reduced by a factor about 8, which is the main advantage of this approach. Evaluations of matching examples show that PCA-SIFT performs slightly worse than standard SIFT-keys [11].

- Local Binary Pattern

Local binary patterns (LBP) are a very simple texture descriptor approach initially proposed by Ojala et al. [12]. They have been used in a lot of applications and are based on a very simple binary coding of thresholded intensity values. In their simplest form they work on a 3 x 3 pixel neighborhood and use the intensity value of the central point as reference for the threshold as shown in Fig. 3.

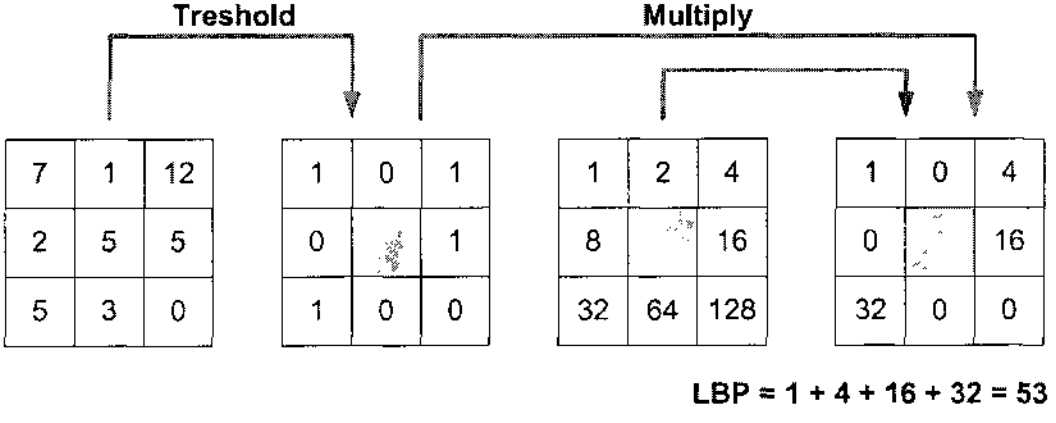


Fig. 3. Illustration of computing LBP feature for 3 x 3 pixel neighborhood. [13]

The neighborhood pixels p_i for $i = 1 \ldots 8$ are then signed (S) according to

$$S(p_0, p_i) = \begin{cases} 1, [I(p_i) - I(p_0)] \geq 0 \\ 0, [I(p_i) - I(p_0)] < 0 \end{cases} \quad (2.2)$$

and form a locally binary pattern descriptor value $LBP(p_0)$ by summing up the signs S, which are weighted by a power of 2 (weight $W(p_i)$). Usually the LBP values of a region are furthermore combined in a LBP-histogram to form a distinctive region descriptor.

$$LBP(p_0) = \sum_{i=1}^{8} W(p_i)S(p_0, p_i) = \sum_{i=1}^{8} 2^{(i-1)}S(p_0, p_i) \quad (2.3)$$

Locally Binary Patterns are invariant to monotonic gray value transformations but they are not inherently rotational invariant. Nevertheless this can be achieved by rotating the neighboring points clockwise so many times, that a maximal number of most significant weight times sign products is zero [12].

2) Global Features

The main idea of global features is to project the original input images onto a suitable lower dimensional subspace. We give a brief overview about some of the important state of the art methods.

- Principal Component Analysis

Principal Component Analysis (PCA) [1] also known as Karhunen-Loeve transformation (KLT), is a well known and widely used technique in statistics. It was first introduced by Pearson [14] and was independently rediscovered by Hotelling [15]. The main idea is to reduce the dimensionality of data while retaining as much information as possible. This is assured by a projection that maximizes the variance but minimizes the mean squared reconstruction error at the same time. Dimension reduction is achieved by

preserving the components that account for the vast majority of the variance in the dataset. The remaining components are discarded. In image recognition, PCA can be used to produce a feature set with a dimensionality significantly less than that of the original images. This reduction is performed without a substantial loss of the intrinsic data contained in the image.

In this process, the 2D training images are converted into N X 1 vectors x_i, where N is the number of pixels in the original image and i ranges from 1 to M images. The next step is to compute the mean image. This is accomplished by summing corresponding positions and dividing the sum by M as shown in equation 2.4.

$$m = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{2.4}$$

Then the mean image m is subtracted from each of the original image vectors. The new mean centered vectors w_i are computed using equation 2.5 and placed in a matrix W.

$$w_i = x_i - m \tag{2.5}$$

The covariance matrix C is computed using equation 2.6.

$$C = WW^T \tag{2.6}$$

The covariance matrix is factored to yield the eigenvectors and eigenvalues. The eigenvectors are used to provide a basis for the projection into the subspace. Additionally, the new basis is oriented in such a fashion that the first eigenvector is associated with the maximum variance. The second eigenvector is the direction with the second greatest variance, and so on.

It is not necessary to calculate the covariance matrix. The eigenvectors can be computed by way of singular value decomposition (SVD) of the image matrix directly. Equation 2.7 shows the factoring where U is the matrix containing the eigenvectors.

$$W = U\Sigma V^T \tag{2.7}$$

PCA is a widely used method for feature extraction. However, it is not suitable to be used as features for rapid face recognition because of its computational overhead. Image data typically has a very high dimensionality. Using this type of data, the PCA feature extraction process always results in a matrix multiplication between two matrices of high dimension, which is computationally expensive.

- Independent Component Analysis

The discussion of Independent Component Analysis (ICA) begins with the meaning of statistical independence. A set of random variables is said to be statistically independent if knowing something about the value of one of the variables does not yield any information about the value of the other. More formally, given a set of random variables $\{y_1, y_2, \ldots, y_m\}$ and a joint density function $f(y_1, y_2, \ldots, y_m)$, the random variables are statistically independent if f can be factorized as follows:

$$f(y_1, y_2, \ldots, y_m) = f(y_1)f(y_2) \ldots f(y_m) \tag{2.8}$$

where $f(y_m)$ is the marginal density of y_m. The goal of ICA is to find a projection such that equation 2.8 holds. Similar to PCA, ICA can be used to reduce the dimensionality of other features.

- Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is another linear transformation method. Unlike PCA, which does not explicitly account for class difference during its analysis process, LDA does. Discriminant analysis, in general, is used to determine which directions discriminate between two or more classes. LDA seeks to find the projection that best separates the dataset by classes. In this case, the best projection is the one that minimizes the variance or scatter between members of the same class and maximizes the separation between different classes. More formally, let $C_1, C_2, \dots, C_m$ be a partitioned dataset of m classes in $\mathbb{R}^n$, then each class C_j has N_j samples $X_i^j \in C_j$ where $i = 1, \dots, N_j$ and $j = 1, \dots, m$. The analysis begins by computing the means of each of the classes using equation 2.9.

$$\mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j} X_i^j \tag{2.9}$$

Next the overall mean is calculated using equation 2.10

$$\mu = \frac{1}{m}\sum_{j=1}^{m} \mu_j \tag{2.10}$$

Then the matrix that represents the separation between classes known as the between-class scatter matrix is computed using equation 2.11

$$S_B = \sum_{j=1}^{m}(\mu_j - \mu)(\mu_j - \mu)^T \tag{2.11}$$

The matrix that captures the separation within the class is known as the within-class scatter matrix and is computed as follows:

$$S_W = \sum_{j=1}^{m} \sum_{i=1}^{N_j} (X_i^j - \mu_j)(X_i^j - \mu_j)^T \tag{2.12}$$

Both S_B and S_W are in $\mathbb{R}^{n \times n}$. The goal is to find a projection such that S_B is maximized and S_W is minimized. In general, a projection of X into new space is accomplished via $Z = U_T X$, where Z is the projection of X in the space spanned by the columns of U. This optimization problem can be solved by applying the general form of the eigenvalue problem:

$$S_B U = \lambda S_W U \tag{2.13}$$

B. Classification

1) Bayesian Decision Theory

Bayesian Decision Theory is a classical method for classification that has wide-reaching applications. It establishes the foundation for making pattern recognition decisions or classifications based on statistical inference. Before continuing it is necessary to state Bayes formula, which will serve as the basis for this discussion. Let $\{\omega_1, \omega_2, \ldots, \omega_C\}$ be a finite set of C states of natures or classes. If x is a feature vector containing observed random variables used to describe the states of nature, then the Bayes formula will be the following:

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)} \tag{2.14}$$

where

$$p(x) = \sum_{j=1}^{C} p(x|\omega_j)P(\omega_j) \tag{2.15}$$

This means that the posterior probability $P(\omega_j|x)$ is equal to the likelihood $p(x|\omega_j)$ times the prior probability $P(\omega_j)$ divided by evidence $p(x)$. The posterior probabilities are computed for all classes and the class with the highest probability is selected. This is the Bayesian Decision rule. By choosing the class with the highest posterior probability the expected error is minimized.

A variation of this rule seeks to minimize the risk associated with making a decision. This is accomplished by adding a loss function to the equation. Let $\{\alpha_1, \alpha_2, \ldots, \alpha_a\}$ be a finite set of a possible actions and let $\lambda(\alpha_i|\omega_j)$ be the loss function associated with selecting action α_i when ω_j is the true state of nature; then the expected loss of action α_i is given by equation:

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)P(\omega_j|x) \qquad (2.16)$$

Using the formulation above, the expected loss or risk associated is computed for all of the possible actions, and the action that yields the minimum risk is selected.

2) K Nearest Neighbor

Unlike the other classifiers, the KNN classifier does not require configuration prior to use except for the determination of k, the number neighbors used, and the presence of the reference set. This simplicity is a major advantage for the KNN classifier over neural networks that require substantial configuration by the researcher.

Classification is a simple matter. The test subject is labeled based on its closest neighbors. The major task is to evaluate the closeness or distance. The Euclidian distance is a standard measure often used. Given a reference point $Y = \{y_1, y_2, \ldots, y_n\}$ and a test

point $X = \{x_1, x_2, \ldots, x_n\}$ where n is the number of random variables describing the points; the Euclidian distance is shown in equation (2.17).

$$d(Y, X) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{2.17}$$

Though the procedure for determining class is relatively simple, the process can yield complicated nonlinear decision boundaries. Unlike the decision boundary of a neural network, the KNN decision boundary is guaranteed to have an expected error no worse than twice that of the optimal solution.

3) Decision Tree

Another tool for classification is the decision tree. The tree structure is used to segregate subjects into the appropriate classes as it traverses the tree. There are three distinct nodes: root, intermediate, and leaf. The root node is the starting point of the process. Using the features of the subject, the root applies a test, which determines to which child the subject should proceed. If the child is a leaf node, the class of the subject is contained in the node and the process is completed. If the child is an intermediate node, the process is repeated using the test that resides at the new node. Ideally, the tests at the higher nodes are more general in nature, and the ones at the lower nodes are more specific and tailored to smaller groups of subjects.

In order for a decision tree to be effective, each node must simplify the problem for its children. Another important consideration is how this simplification process is measured. Impurity is the metric that is traditionally used. There are different mathematical formulations for impurity. Equation 2.18 shows the most popular [16].

$$i(N) = -\sum_j P(\omega_j) \log(\omega_j) \tag{2.18}$$

$i(N)$ denotes the impurity at node N. The simplification can be monitored by measuring the change in impurity between the parent node and its children nodes. For a tree that is restricted to binary splits, the change in impurity is given by

$$\Delta i(N) = i(N) - P_L i(N_L) - P_R i(N_R) \tag{2.19}$$

where $N_L$and N_R are the left and right children nodes respectively, and P_L and P_R are the fraction of the parent's dataset in each child node. If multiway splits are allowed the equation for $\Delta i(N)$ changes to

$$\Delta i(s) = i(N) - \sum_{k=1}^{B} P_k i(N_k) \tag{2.20}$$

where B is the number of children in the split, P_k is the fraction at node k, and $i(N_k)$ is the impurity at node k. This equation can be used to determine the size of the optimal split.

III. ROBUST FACE RECOGNITION SYSTEM

In this chapter we present the contributions made in improving the existing methods which are used to achieve robust face recognition. To aid faster and more accurate face detection we developed a new skin segmentation method. We use an existing face detection method, which uses Haar features and AdaBoost classifier to detect faces. We developed a face color constancy method which will be used in the preprocessing stage. This method transforms the face images towards reference illumination condition which will give our system some illumination invariance. We propose two methods to perform face recognition using modular PCA and boosted tree of classifiers. In the following sections, we describe in detail the contributions made towards pose, illumination and expression invariant face recognition.

A. Face Recognition System

Our proposed face recognition system takes still images or video as input. The skin segmentation method is applied to segment the regions in images which have skin like color. Face detection is applied on the segmented regions. Since face detection is limited to skin color regions, we can get more accurate face detection. Next the detected faces are processed using color constancy method. This transforms the face illumination to reference illumination condition. Next, we perform face recognition to classify the face. Fig. 4 shows the blocks of our face recognition system.

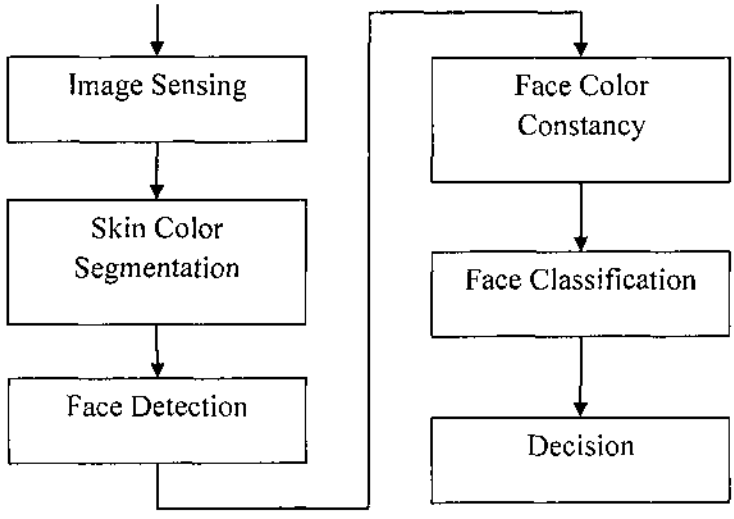


Fig. 4. Face recognition system.

B. Skin Color Segmentation

Skin segmentation can be defined as the process of identifying the pixels of a given color image which corresponds to human skin. Humans use color as a fundamental clue for detecting faces from complex scenes. Skin color segmentation is a difficult problem since skin colors vary based on the ambient light and the cameras, which produce different colors even for the same person under the same illumination conditions. Skin colors also vary from person to person.

Probabilistic methods have been widely used for pixel based skin segmentation. One of the probabilistic methods is the Skin Probability Map (SPM) [18, 19], which has been found to be the best in terms of accuracy and running time [19]. Other well known statistical skin color models are single Gaussian model [20] and the mixture of Gaussian model [21]. The single Gaussian skin model has the advantage of being simple and fast, it however does not adequately represent the variance of the skin distributions under

different conditions. To overcome this drawback, the mixture of the Gaussian model has been suggested. It is however hard to be trained and slow. Some of the earlier skin segmentation techniques classify pixels based on pre-defined ranges in the color space [23]. More recently machine-learning algorithms have been used to perform pixel classification based on predefined ranges in color space [22]. The method we use for skin modeling is SNoW (Sparse Network of Winnows). It was first used in image processing for face detection with great success [24]. We consider the skin pixels and non-skin pixels as two classes and perform the classification using SNoW.

The choice of color space has a significant impact on the skin segmentation result. Researchers have tried to find the most effective color space for skin segmentation by performing detailed analysis on different color spaces [27, 28]. But there is no single color space, which is most effective for all skin segmentation methods [25]. It has been argued theoretically that for every color space, there exists an optimum skin segmentation scheme such that the performance of all these skin segmentation schemes is the same in that color space [26].

1) Skin Color Model Using SNoW

We trained the SNoW algorithm on the YCbCr color space since the transformation from RGB is simple and it has explicit separation of luminance and chrominance components. We say more about the choice of color space in the next section. The network is first trained in the YCbCr color space with the image created from the skin patches. A two-dimensional skin distribution matrix (SDM) of dimensions 256×256 is created and all its elements are initialized to 0. During training with the skin patches, the

elements of the SDM are selected for updating based on the index computed as $[Cb(x,y), Cr(x,y)]$. The winnows update rule is used to update the matrix and the update procedure is as shown:

$$if\ W\big(Cb(x,y), Cr(x,y)\big) == 0 \qquad \forall x,y$$

$$W\big(Cb(x,y), Cr(x,y)\big) = \eta \tag{3.1}$$

The elements or weights of the SDM which are active are initialized to η as shown in equation (3.1); η is 0.1 in our implementation. Next the weights are promoted for all the pixels in the skin image as shown in equation (3.2).

$$if\ W\big(Cb(x,y), Cr(x,y)\big) < \eta \times \alpha^{\eta} \quad \forall x,y$$

$$W\big(Cb(x,y), Cr(x,y)\big) = \alpha \times W\big(Cb(x,y), Cr(x,y)\big) \tag{3.2}$$

where W is the SDM; Cb, Cr represent the color components of the skin image and x, y vary from 1 to X, Y respectively where X is the number of columns in the skin image and Y is the number of rows in the skin image. The variable α in equation (23) is used for promoting the weights, and α=2 in our case. The result of initial skin segmentation performed on a test image and the original image are shown in Fig. 5. Skin segmentation is performed by classifying pixels with weights above 1000 (selected after experimental analysis) as skin pixels. In Fig. 5, pixels classified as skin are represented as white and non-skin pixels are represented as black. As can be seen in Fig. 5, many non-skin pixels are falsely classified as skin pixels. Non-skin patches falsely classified as skin pixels are used as negative examples to update the SDM as shown in equation (24), where t = 1000 and β=0.5.

$$if\ W\big(Cb(x,y),Cr(x,y)\big) > t\ \forall x,y$$

$$W\big(Cb(x,y),Cr(x,y)\big) = \beta \times W\big(Cb(x,y),Cr(x,y)\big) \tag{3.3}$$

More negative examples of non-skin pixels are obtained by performing skin segmentation on test images and the non-skin patches are incorporated into the non-skin image This process is performed until the false positives are reduced to a desired level. Equation (3.3) performs the demotion of weights for pixels from the non-skin regions, which overlap with the skin region in the Cb-Cr color space. The advantage of using the winnow update rule for updating the weights is that we can reduce the false positives without adversely affecting the true positive rate. The SDM in Cb-Cr color space after updating it with non-skin pixels is shown in Fig. 6.

Fig 5. Selecting non-skin patches.

The SDM can be normalized by dividing it with the maximum weight in the SDM The normalized SDM would give the probability of skin distribution not overlapping with the

non-skin distribution. Unless stated otherwise, SDM refers to a non-normalized distribution.

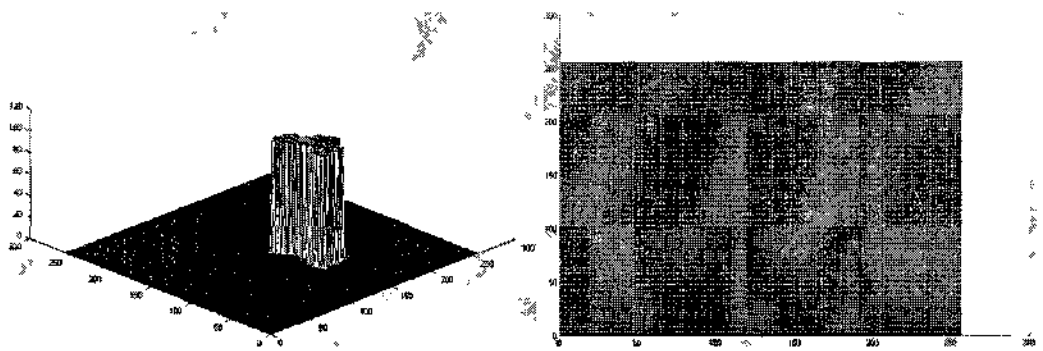


Fig. 6. SDM in Cb-Cr after training.

2) Selecting Color Spaces

We are interested in selecting a color space, which would give the best skin segmentation results for our technique. From the literature we found that one of the best skin segmentation was obtained using the TSL color space [27]. We trained the SDM on the TSL color space and YCbCr color space. The results obtained are as shown in Fig. 7.

Fig. 7. Original images used for testing are shown in the first column, second column shows the results of skin segmentation using SDM in Cb-Cr color space, and results of skin segmentation using SDM in T-S color space are shown in column three.

C. *Color Constancy*

We propose a novel technique to compensate for variations in face images due to changes in the illumination conditions. Other than color of the faces, shadows and specular reflections on the faces also change with respect to lighting environment. These variations result in false recognitions even in the best face recognition algorithms. We

tackle this problem by processing the faces using a color constancy model specific to face images, which we named face color constancy (FCC).

Our approach of using color constancy to achieve illumination invariant face recognition is unique in many ways compared to previous illumination invariant face recognition methods. We can deal with real world illumination environments where many light sources are active simultaneously, the position of the light sources are changing and the neighborhood scene is changing. Our model is also capable of FCC when camera processing such as auto-gain and camera color balancing functions are active. Given a few faces under different illumination environments in these real word situations, our model can learn the basis vectors needed to compensate for these variations in any new face image. Our model can also compensate for strong shadows and specular reflections to some extent. We process the RGB values of the face image and aim to minimize the RMS error in each band with respect to the reference face in the gallery. Processing on the RGB image gives us more power to minimize the RMS errors. We can also support a face recognition system where each subject in the gallery is in a different lighting environment. A probe image taken under an arbitrary illumination environment can then be transformed towards its reference in the gallery using our FCC algorithm.

1) Learning the Joint Color Changes

Let the RGB color space be defined as, $C = \{(r, g, b)^T \in \mathbb{R}^3 : 0 \leq r \leq 255, 0 \leq g \leq 255, 0 \leq b \leq 255\}$. This space defines all the possible color vectors observable in images. The color vector of an image pixel p is denoted as $c(p) \in C$. Let $1 \leq i \leq N$, where N is the number of individuals and let $1 \leq j \leq M$, where M is the number of

illumination conditions under which each individual's image is taken. Also let $1 \leq k \leq P$, where P is the number of pixels in each face image. The mapping of colors under different illumination conditions is represented by difference of two corresponding pixels:

$$d\left(I_{i0}^{k}, I_{ij}^{k}\right) = c\left(I_{i0}^{k}\right) - c\left(I_{ij}^{k}\right) \tag{3.4}$$

This difference vector tells us how a particular pixel's value changed from illumination condition of I_{i0} to illumination condition of I_{ij}. This is computed for each of the P pair of pixels to obtain a vector field that is defined at all points in C for which there are colors in image I_{ij}. The vector field is constructed by placing each vector difference at the point $c(I_{ij}^{k})$ in the color space C. The vector field Φ' over C is defined as:

$$\Phi'\left(c\left(I_{ij}^{k}\right)\right) = d\left(I_{i0}^{k}, I_{ij}^{k}\right), \quad 1 \leq k \leq P \tag{3.5}$$

This vector field is only defined at particular color points in C that happen to be in image I_{ij}^{k}, hence Φ' is called a partially observed color flow. We wish to obtain a full color flow from the partially observed color flow. A simple approach to obtain the full color flow is to follow an interpolation scheme as proposed in [17]. The color flow at a color point $(r, g, b)^{T}$ is obtained by a weighted proximity based average of nearby observed color flow vectors.

$$\Phi(r, g, b) = \frac{\sum_{k=1}^{P} e^{-\left\| c\left(I_{ij}^{k}\right) - (r,g,b)^{T} \right\|^{2} / 2\sigma^{2}} \Phi'\left(c\left(I_{ij}^{k}\right)\right)}{\sum_{k=1}^{P} e^{-\left\| c\left(I_{ij}^{k}\right) - (r,g,b)^{T} \right\|^{2} / 2\sigma^{2}}} \tag{3.6}$$

In [17] the full color flow is defined at every point in C, whereas our full color flow is defined only in a subset of C. The points where Φ is defined depends on the colors present in I_{ij}. The variance term σ^2 controls the mixing of the observed flows to form the interpolated flow vectors. Visualization of the color flow vector field for a face pair is shown in Fig. 8.

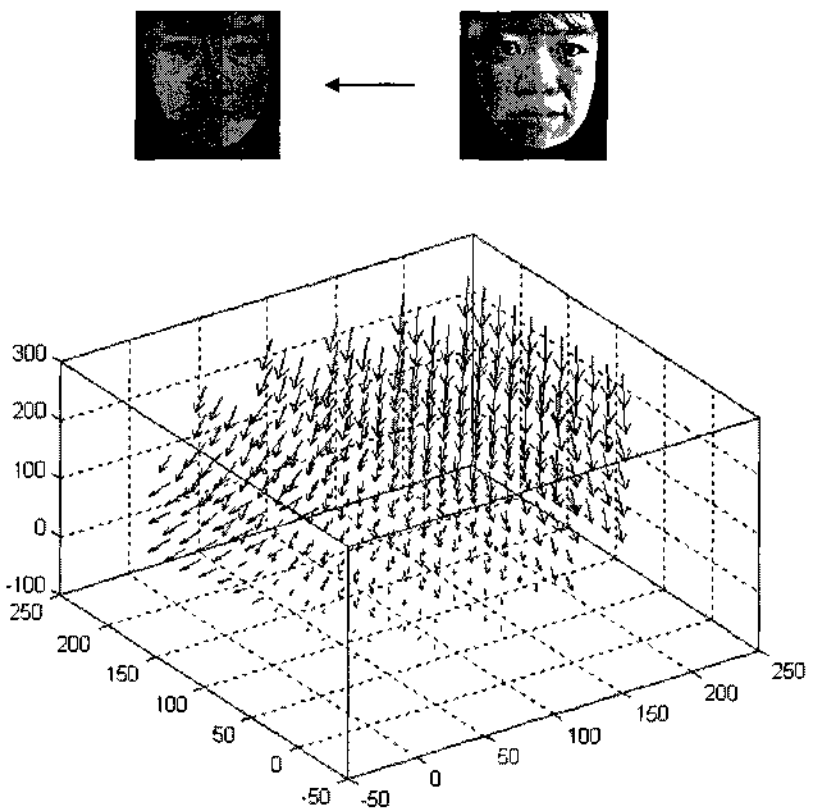


Fig. 8. Visualization of the full color flow vector field.

The possible changes of a pixel's color on the face surface due to changes in illumination conditions are compact. While in principle its possible for a change in illumination condition to map any color from a Lambertian surface to any other color independently of all other colors, we know from experience that many such joint maps

are not observed in real world situations. Hence there is significant structure in the space of color flows. The image pair shown in Fig. 8 has a nonlinear variation; some regions of the face become brighter than other regions due to position of the point source. As can be seen from the visualization, this kind of nonlinear variations can be captured by modeling the space of color flows.

Given a large number of color flows, we wish to model their distribution. We chose to use Principal Component Analysis (PCA) due to the following reasons. The flows are well represented by a small number of principle components and finding the optimal description of a difference image in terms of color flows is computationally efficient using this representation. There are ~16 million points in the color space C, hence to represent the color flows we quantize it at Q distinct points. Therefore the color flow Φ can be represented as a collection of $3Q$ coordinates. We chose $Q = 4096$ distinct and equally spaced points in the color space for our experiments. Hence the full color flow is a vector of 3×4096 components. Using a higher value of Q would give us a more accurate color flow field, but due to computational speed and memory limitations we settle for $Q = 16^3$. We compute principal components of the color flow covariance matrix. These principal components are known as eigenflows [17]. Fig. 9 shows the eigenvalues associated with the first 50 eigenflows. As expected the curve drops off rapidly indicating that most of the variance in the color flow distribution is represented by the first few eigenflows.

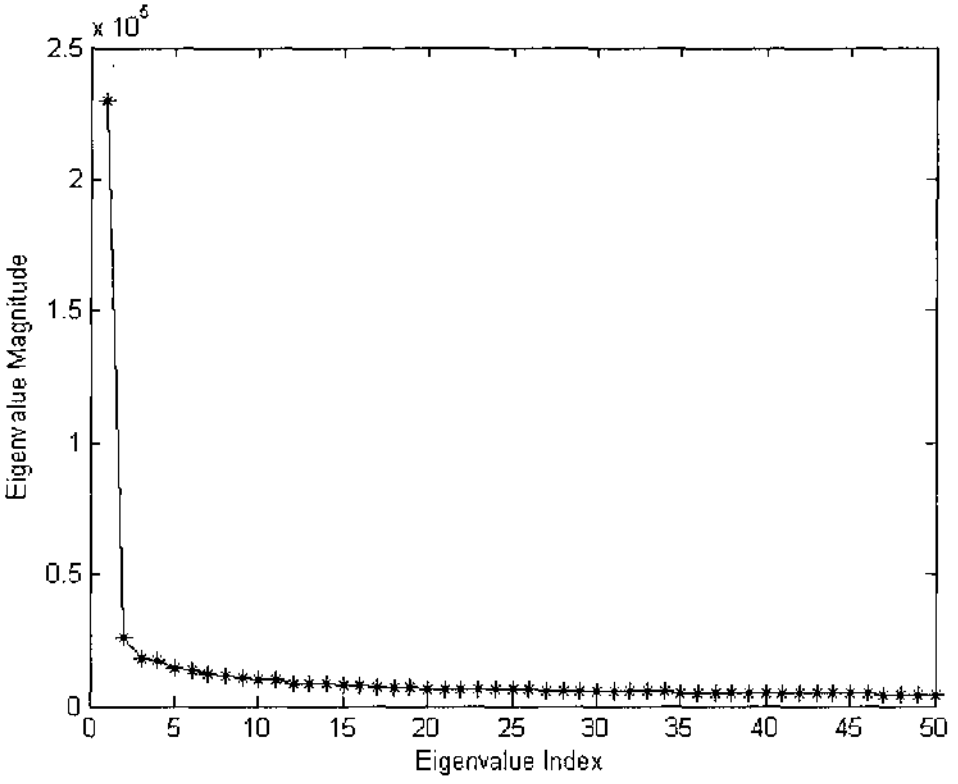


Fig. 9. Magnitude of the eigenvalues vs. eigenvalue index.

2) Color Constancy from Eigenflows

Given an image taken under arbitrary illumination condition, how can we obtain the image closest to its reference image in terms of L2 distance? In previous section we described the training procedure to obtain the eigenflows for face images. Using these eigenflows, we can transform a face image taken under arbitrary illumination conditions towards its reference illumination condition.

Let I_{test} be a face under arbitrary illumination condition and let I_{test0} be its reference. We compute the difference image as

$$D = I_{test0} - I_{test} \tag{3.7}$$

The difference image basis vectors for the test image and a set of E eigenflows Ψ_i, $1 \leq i \leq E$, can be represented as

$$D_i = I_{test}(\Psi_i) \tag{3.8}$$

Here the operator $I_{test}(.)$ takes the pixel values at the location $[x, y]$ and generates a difference image basis vector by placing at each $[x, y]$ the closest eigenflow. The closest eigenflow is determined based on the distance in the color space from the color vector at $[x, y]$. The transformed image is obtained as

$$I_T = I_{test} + \sum_{i=1}^{E} \gamma_i D_i \tag{3.9}$$

where γ_i are scalar multipliers. We can directly solve for γ_i by solving the system

$$\gamma_i = D_i^{\dagger} D \tag{3.10}$$

Here D is the difference image defined in equation (3.7), and $D_i^{\dagger}$ is the pseudo-inverse of the difference image basis vectors defined in equation (3.8).

D. Modular Face Recognition

The main objective of this research is to improve the accuracy of face recognition subjected to varying head pose, illumination and facial expression. As stated before, PCA method has been a popular technique in facial image recognition. But this technique is not accurate when the pose and illumination of the facial images vary considerably. In this research work, we improve the accuracy of this technique under these conditions. We propose the modular PCA method, which is an extension of the conventional PCA

method. In the modular PCA method the face images are divided into smaller images and the PCA method is applied on each of them. Whereas in the traditional PCA method the entire face image is considered, hence any profound variation in pose or illumination will affect the recognition rate profoundly. Since in the case of modular PCA method the original face image is divided into sub-images the variations in pose or illumination in the image will affect only some of the sub-images, hence we expect this method to have better recognition rates than the conventional PCA method. A similar method called modular eigenspaces was proposed by Pentland et al., 1994. In this method PCA is performed on the eyes and nose of the face image.

The PCA based face recognition method is not very effective under the conditions of varying pose and illumination, since it considers the global information of each face image and represents them with a feature set. Under these conditions, the features will vary considerably from the features of the images with normal pose and illumination, making it difficult to classify them correctly. On the other hand, if the face images were divided into smaller regions and the features are computed for each of these regions, then the features will be more representative of the local information of the face. When there is a variation in the pose or illumination, only some of the face regions will vary and rest of the regions will remain the same as the face regions of a normal image. Hence, features of the face regions not affected by varying pose and illumination will closely match with the features of the same individuals face regions under normal conditions. Therefore, it is expected that improved recognition rates can be obtained by following the modular PCA approach. We expect that if the face images are divided into very small

regions the global information of the face may be lost and the accuracy of this method will deteriorate.

1) Modular PCA

In this method, each image in the training set is divided into N smaller images. Hence the size of each sub-image will be L^2/N. These sub-images can be represented as

$$I_{ij}(m,n) = I_i\left(\frac{L}{\sqrt{N}}(j-1)+m, \frac{L}{\sqrt{N}}(j-1)+n\right) \quad \forall i,j \tag{3.11}$$

where i varies from 1 to M, M being the number of images in the training set, j varies from 1 to N, N being the number of sub-images and m and n vary from 1 to $L/\sqrt{N}$. The average image of all the training sub-images is computed as

$$A = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij} \tag{3.12}$$

The next step is to normalize each training sub-image by subtracting it from the mean as

$$C = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} Y_{ij}Y_{ij}^T \tag{3.13}$$

Next we find the eigenvectors of C that are associated with the M' largest eigenvalues. We represent the eigenvectors as $E_1, E_2, \ldots E_{M'}$. The features are computed from the eigenvectors as shown below:

$$W_{pnjK} = E_K^T \times \left(I_{pnj} - A\right) \quad \forall p,n,j,K \tag{3.14}$$

where K takes the values $1,2\ldots M'$, n varies from 1 to Γ, Γ being the number of images per individuals, and p varies from 1 to P, P being the number of individuals in the training

set. Features are also computed for the test sub-images using the eigenvectors as shown in the next equation:

$$W_{test\ jK} = E_K^T \times \left(I_{test\ j} - A\right) \quad \forall j, K \tag{3.15}$$

Mean feature vector of each class in the training set is computed from the feature sets of the class as shown below:

$$T_{pjK} = \frac{1}{\Gamma}\sum_{K=1}^{M'}\sum_{n=1}^{\Gamma} W_{pnjK} \quad \forall p, j \tag{3.16}$$

Next the minimum distance is computed as show below:

$$D_{pj} = \frac{1}{M}\sum_{K=1}^{M'}\left|W_{test\ jK} - T_{pjK}\right| \tag{3.17}$$

$$D_p = \frac{1}{N}\sum_{j=1}^{N} D_{pj} \tag{3.18}$$

$D_p < \theta_i$ for a certain value of p, the corresponding face class in the training set is the closest to the test image. Hence the test image is classified as belonging to the p^{th} face class.

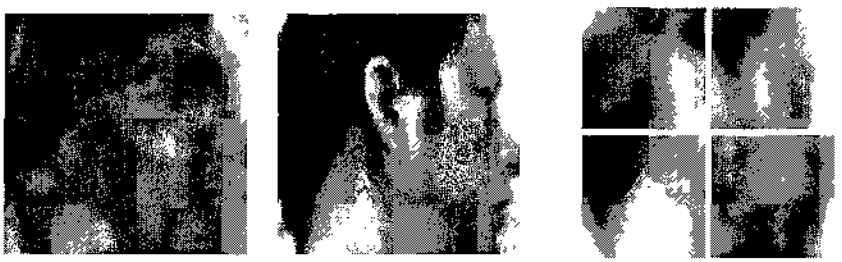
2) Comparison

We applied the PCA method and the modular PCA method to reconstruct the test images. In the case of the PCA method the image is reconstructed as

$$I_{test} = A + E_K^T W_{test\ K} \tag{3.19}$$

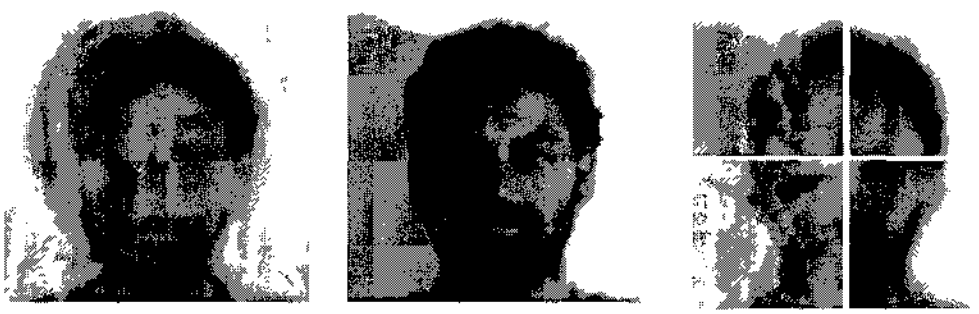
The test image is reconstructed in a similar manner for the modular PCA method and is given as

$$I_{test\ j} = A + E_K^T W_{test\ jK} \tag{3.20}$$



(a) (b) (c)

Fig. 10. (a) Reconstructed image using PCA method for a test image from the Sheffield database, (b) Original image from the Sheffield database, and (c) Reconstructed image using modular PCA method at N=4 for a test image from the Sheffield database.



(a) (b) (c)

Fig. 11. Reconstructed image using PCA method for a test image from the Yale database, (b) Original image from the Yale database, and (c) Reconstructed image using modular PCA method at N=4 for a test image from the Yale database.

Figures 10 and 11 show the reconstructed images of a face image from the test set of the Sheffield [29] and Yale [30] databases using both the methods. In the figures, the

reconstructed images obtained for the modular PCA method are concatenated to facilitate visual comparison with the reconstructed image obtained for PCA method

IV. FACE RECOGNITION USING BOOSTED TREE

We build a tree of discriminative classifiers by learning the intra-personal and extra-personal data recursively. At the root level a strong classifier is trained, we then split the training data based on the posterior probability of this classifier. Training data which is classified with high probability as intra-personal variations is sent to the right set. Similarly training data which is classified with high probability as extra-personal variation is sent to the left set. The remaining data is sent to both the sets. The right child and left child strong classifiers are then trained on these sets. This process is repeated on each child strong classifier until the stopping conditions are met. By this process the child strong classifiers will learn the features needed to classify the difficult data which was classified with low probability or misclassified by its parent.

A. Related Work

One of the influential algorithms for face recognition was proposed by Moghaddam and Pentland [33]. They model the distribution of intra-personal and extra-personal space. To model the distributions they use eigenspace density estimation and obtain the probability density functions of both these classes. Bayesian analysis is used to obtain the similarity measure for test images. The key advantage of this method is that it learns to distinguish between variations in images of same subjects against variations between different subjects. Jones and Viola [34] learn a boosted classifier on these difference images using AdaBoost and Haar like features. They show that a very accurate classifier can be learned by learning a few important features. They also propose a resampling

method to deal with the large number of extra-personal data compared to the small number of intra-personal data. In paper [35] Yang et al. combined the ideas from [33] and [34] with Gabor features to show that Gabor features selected by AdaBoost can give high recognition rates.

In [36] a probabilistic boosting tree (PBT) framework is proposed to learn two-class and multi-class discriminative models. This method recursively constructs a tree of strong classifiers with the training data. This is done by dividing the training set to two new sets, left and right set. Each of the sets is then used to train the left and right sub-trees recursively. Hard to classify data is sent to lower levels, leading to expansion of the tree. One of the key advantages is that clusters in data are automatically discovered. Once trained the tree can be used to compute the posterior probability of given input data. This framework was used successfully for multi-class object recognition and multi-view face detection. A related approach called cluster boosted tree (CBT) is presented in [37]. This method learns tree structured classifiers by the dividing the training data using discriminative image features. It uses vector boosted tree [39] to build the classification tree. At each boosting round one weak classifier is selected for each branch. If the discriminative power of the new weak classifier is too weak the branch is split in two. When a branch splits, the two resulting branches are re-trained from the beginning. This is done to propagate the sub-categorization information upstream. At the end of training, each branch forms a strong classifier for the sub-categories discovered automatically from the training data. The key improvement of CBT over PBT is that a balanced splitting of positive and negative data is achieved. This method was applied to pedestrian

detection and car detection and was shown to be more accurate than view-based detectors like [39].

B. Motivation for Our Approach

We are interested in learning the features which can separate intra-personal and extra-personal classes accurately. As shown in [34], a linear combination of rectangle features can be learned to discriminate between these two classes. Building on this idea we use Gabor features in our approach. Gabor features have been successfully used before in achieving high accuracy in face recognition [38].

The intra-personal space is made up of variations caused due to external factors like illumination, expression, pose, occlusion and noise. Whereas the extra-personal space will have variations due to facial features between different subjects in addition to the above mentioned factors. These external factors will cause some intra-personal and extra-personal data to be intermixed in the feature space. One possible way to decouple these two classes is to design more effective features. Unfortunately it is often hard to design features which can give a good separation between these two classes in the feature space.

Our approach is to learn a small set of features from the original high dimensional Gabor feature space using a tree of strong classifiers learned using AdaBoost. In AdaBoost each round of boosting selects a weak classifier (feature) with minimum error rate. The samples which are incorrectly classified are given more weight and the weights of all the samples are normalized. Due to this previously correctly classified samples may be mis-classified again and receive penalty. Thus, after some rounds of boosting weak classifier become ineffective. Instead of putting all the weak classifiers in a single strong

classifier we use the divide and conquer approach to build a tree of strong classifiers. This gives us a better chance of learning the features to separate the hard to classify samples deeper down the tree.

To illustrate the working of our method we trained it with a subset of data from AR face database [31]. This subset had 30 randomly selected subjects and each subject has 13

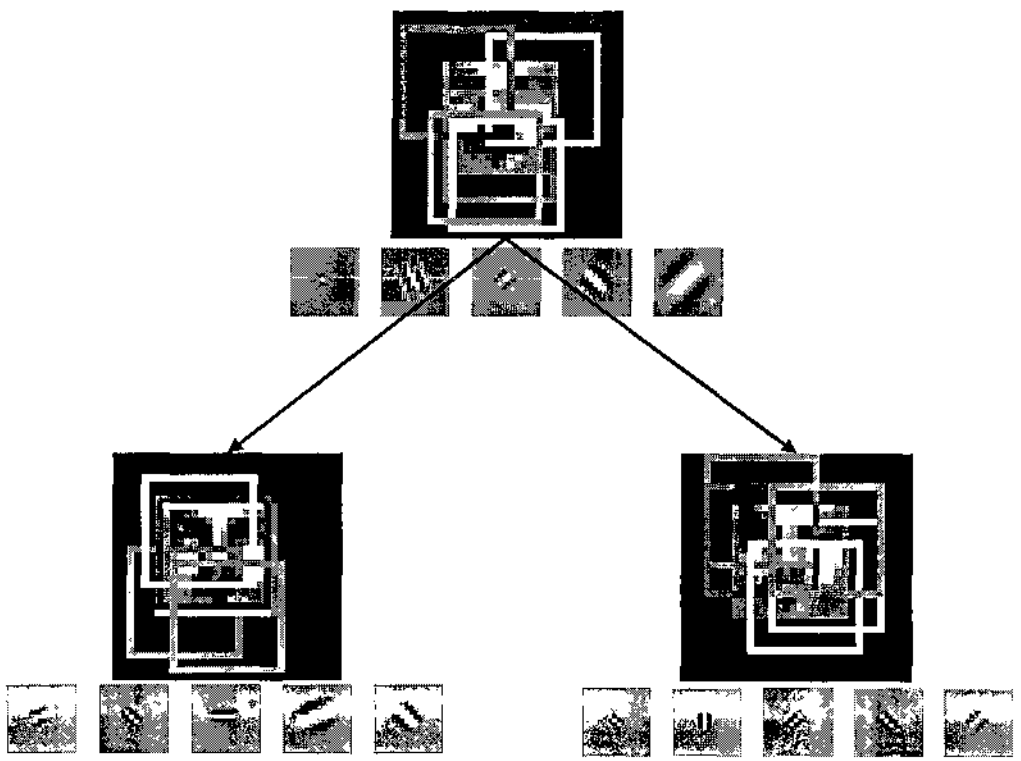


Fig. 12 The root and first level of the learned classifier tree with top five feature locations overlaid on the face image. The order of top features in decreasing order of importance is colored red, green, blue, yellow and pink. The corresponding Gabor filters are displayed below the image.

images. These images have different illumination, expression and occlusion (sun glasses and scarf) as shown in Fig. 24. The top 5 features learned by the root classifier and right and left child classifiers as shown in Fig. 12. The right classifier was automatically trained with data consisting of intra-personal samples and extra-personal samples which were classified with low probability by the root classifier. This classifier naturally focuses on learning features to correctly classify the hard to classify extra-personal samples against the intra-personal samples. Similarly the left classifier has learned the features need to classify the hard to classify intra-personal samples.

C. Gabor Features on Difference Images

Gabor features capture the salient visual properties. They have desirable characteristics such as spatial localization, orientation selectivity, and spatial frequency. They model the receptive field profiles in cortical simple cells. We use Gabor filters to extract features from the difference images. Using difference images reduces the multi class problem to two class problem. It also facilitates the use of binary classifiers like AdaBoost.

1) Intra-personal and extra-personal differences

In this paper we consider face recognition as the problem of determining if a pair of face images belongs to same subject or different subjects. To do this we learn the Gabor features from intra-personal and extra-personal difference images. Gabor features are to some extent insensitive to variations like illumination and expression. However its ability to decouple intra-personal and extra-personal variations is limited.

Give a training set of images with K subjects and M images per subject, the amount of intra-personal and extra-personal data would be $N^+ = KM(M-1)/2$ and $N^- = M^2K(K-1)/2$ respectively. The ratio of number of extra-personal data to number of intra-personal data would be $M(K-1)/(M-1)$. Typical face recognition problems we are interested in have very large number of subjects compared to number of image per subject. Hence the number of extra-personal data is very large compared to intra-personal data. To deal with this large asymmetry in training data, resampling method proposed in [34] is used in training the AdaBoost classifier.

2) Gabor features

Gabor kernel is the product of a Gaussian envelope and a plane wave, defined as

$$\psi_{\vec{k}}(\vec{x}) = \frac{\|\vec{k}\|^2}{\delta^2} e^{(-\|\vec{k}\|^2 \|\vec{x}\|^2 / 2\delta^2)} \left[e^{i\vec{k}\vec{x}} - e^{-\delta^2/2} \right] \tag{4.1}$$

where $\vec{x} = (x, y)$ is the variable in spatial domain. Here $\vec{k}$ is the frequency vector which determines the scale and orientation of the Gabor kernels, and is defined as

$$\vec{k} = k_s e^{i\phi_d} \tag{4.2}$$

where $k_s = k_{max}/f^s$ and $\phi_d = \pi d/8$ with k_{max} being the maximum frequency, and f is the spacing factor between kernels in the frequency domain. In face recognition researchers commonly use 40 Gabor wavelets with five scales $s \in \{0,1,2,3,4\}$ and eight orientations $d \in \{0,1,2,3,4,5,6,7\}$. Also $\delta = 2\pi$, $k_{max} = \pi/2$, and $f = \sqrt{2}$.

Given an image $I(\vec{x})$ having n^2 pixels, its Gabor transformation at a particular position $\vec{x}_0$ is computed by convolving with Gabor kernels as

$$(\psi_{\vec{k}} * I)(\vec{x}_0) = \int \psi_{\vec{k}}(\vec{x}_0 - \vec{x}) I(\vec{x}) d^2(\vec{x}) \tag{4.3}$$

In this way for each pixel position in the face image 40 complex values are calculated. Phase information of the transform is time-varying hence only the magnitude values are used. This will result in a feature space of dimension $n^2 \times 40$. From this high dimensional feature space we will be selecting a small number of features which minimize classification error using our learning algorithm.

D. Learning Algorithm

In our learning algorithm we use AdaBoost invented by Freund and Schapire [40] to learn the strong classifiers. Strong classifier are created by combining a set of weak classifiers $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ where $h_t(x)$ is a weak classifier. To learn the boosted tree we use the probabilistic boosting tree (PBT) framework proposed in [36].

1) Training Tree of Classifiers

From the training data a strong classifier is trained using AdaBoost. The posterior probability of the training data is computed with the strong classifier as

$$q(+1|x) = \frac{\exp\{2H(x)\}}{1+\exp\{2H(x)\}} \tag{4.4}$$

$$q(-1|x) = \frac{\exp\{-2H(x)\}}{1+\exp\{-2H(x)\}} \tag{4.5}$$

The training data is then split into right and left sets using these probabilities. Samples whose $q(+1|x)$ probabilities fall above $\frac{1}{2} - \varepsilon$ are sent to the right data set and sample whose $q(-1|x)$ probabilities fall above $\frac{1}{2} - \varepsilon$ are sent to the left data set. The remaining

a. Given training set of samples $S = (x_1, y_1, d_1), \ldots, (x_n, y_n, d_n)$, where $x_i \in X$, $y_i \in \{-1, +1\}$ for negative and positive examples respectively and $\sum_i d_i = 1$.

b. Initialize weights $d_{1,i} = \frac{1}{N}$ where N is the total number of negative and positive samples.

c. On the training set S train a strong classifier using AdaBoost with T weak classifiers but exit early if $\epsilon_t > \theta$.

d. While training strong classifier perform resampling for the first round of boosting and every R^{th} round thereafter.

e. If the current tree depth is L then exit.

f. Initialize two empty sets S_{left} and S_{right}.

g. For each sample is S compute the probabilities $q(+1|x_i)$ and $q(-1|x_i)$ using the current strong classifier.

h. If $q(+1|x_i) - \frac{1}{2} > \epsilon$ then $(x_i, y_i, 1) \rightarrow S_{right}$

else if $q(-1|x_i) - \frac{1}{2} > \epsilon$ then $(x_i, y_i, 1) \rightarrow S_{left}$

else $(x_i, y_i, q(+1|x_i)) \rightarrow S_{right}$ and $(x_i, y_i, q(-1|x_i)) \rightarrow S_{left}$.

i. If $min\,(N_{left}^{-1}, N_{left}^{+1})/max\,(N_{left}^{-1}, N_{left}^{+1}) < \varphi$ then exit, where N_{left}^{-1} is the number of samples in S_{left} belonging to class -1 and N_{left}^{+1} is the number of samples in S_{left} belonging to class $+1$.

j. Normalize all the weights of the samples in S_{left}.

k. Repeat the above procedure from step c with S_{left}.

l. If $min\,(N_{right}^{-1}, N_{right}^{+1})/max\,(N_{right}^{-1}, N_{right}^{+1}) < \varphi$ then exit, where N_{right}^{-1} is the number of samples in S_{right} belonging to class -1 and N_{right}^{+1} is the number of samples in S_{right} belonging to class $+1$.

m. Normalize all the weights of the samples in S_{right}.

Fig. 13. Training procedure for boosted tree of classifiers.

samples are the hard to classify samples and they are sent to both the data sets. The importance weights of hard to classify samples are assigned as $q(+1|x)$ for right set and $q(-1|x)$ for left set. The remaining samples are given weights of 1. This ensures that the

child classifiers will not over fit to classify the hard samples correctly. New strong classifiers are then trained on both the sets. This process is repeated recursively on the right and left branches until stopping conditions are met. The variable ε is used to control the over fitting to some extent. After training is completed each node of the tree is a strong classifier.

After each split we count the number of samples from positive and negative classes in the new data set. If the number of samples from one class is below a certain ratio with respect to the other class we stop training on that branch. We stop training the strong classifiers when the error of a weak classifier is above θ or when the boosting has run for T rounds. Fig. 13 gives the procedure we use for training. In all our experiments $\theta = 0.45$, $R = 10$, $\epsilon = 0.1$ and $\varphi = 0.01$.

To test an input difference image, we compute its Gabor features and classify it recursively using the learned tree as described in [36].

2) Resampling

As we have mentioned before the ratio of number of extra-personal data to number of intra-personal data would be $M(K-1)/(M-1)$. Here K is the number of subjects and M is number of images per subject. For a typical case where M is 10 and K is 100, we would have 110 times more extra-personal data compared to the amount of intra-personal data. Having overwhelmingly more number of negative samples over number of positive samples can make it difficult in selecting good features for AdaBoost. To handle this problem we choose limited number of samples from the large number of negative samples using the resampling approach similar to the one used in [34].

We fix the number of positive samples to be 15% of the number of negative samples in the resampled set. We also fix the maximum number of positive samples in the resampled set to be 300. To select the samples we compute the cumulative distribution of the weights of all the positive and negative samples.

$$C_k^{+1} = \sum_i d_i^{+1};\ C_k^{-1} = \sum_i d_i^{-1} \tag{4.6}$$

Let N^{+1} be the number of positive samples and N^{-1} be the number of negative samples we want in the resampled set. Generate N^{+1} uniformly distributed random numbers r_s in the interval [0, C_M^{+1}], where C_M^{+1} is the total weight of all the positive samples. Choose sample x_j^{+1} given the number r_s such that $C_j^{+1} < r_s < C_{j+1}^{+1}$. Similar process is done to select N^{-1} negative samples for the resampled set.

V. EXPERIMENTAL RESULTS

In this chapter we present the experimental results of completed research. To perform the experiments we used several standard data sets. The details of the data sets are given in table 1.

Table 1. Datasets used to perform experiments on the proposed methods.

Task	Dataset	Classes	Images
Skin Segmentation	AR [31]	126	~4000
Face Color Constancy	PIE [32]	68	41368
Face Recognition	Sheffield [29]	20	564
	Yale [30]	15	165

The AR face database has face images taken under different lighting conditions. We use these to collect positive example to train the SDM for skin color segmentation. PIE database has images taken with different head poses, positions of light sources, and facial expressions. We are interested in using the images with different light source positions to train and test our FCC algorithm. The Sheffield face database has images with different head poses varying from profile to frontal. We use this database to test the performance of face recognition algorithms under varying head pose. The Yale face database had images with simultaneous lighting and expression variations. This is use to test performance of face recognition algorithms under varying lighting and expression.

A. Skin Color Segmentation

We created a set of skin pixels from images in the AR face database [29] and personal images taken from a digital camera. The images have illuminations varying from fluorescent light to direct sunlight. From the outdoor scenes we used skin regions from both direct sunshine and shadows. We didn't use any images from the internet during training since there is no control on the image properties. Training was done on the *Cb-Cr* color space since the transformation from RGB is simple and it has explicit separation of luminance and chrominance components.

1) Comparative Evaluation

We evaluate the performances of SDM based skin segmentation scheme and the SPM based skin segmentation method under the same conditions. Both the methods were trained with the same skin and non-skin pixels and tested on a set of images collected from different sources. The constants and thresholds used in the algorithms are not changed across the two color spaces.

We compared the true positives (*TP*) and false positives (*FP*) of both the methods for the test images. The results shown in table 2 were generated by counting the number of pixels classified as skin and non-skin for the first original image shown in Fig. 18 to obtain the *TP* and *FP*. The results are for 4983 skin pixels and 191625 non-skin pixels from the test image.

The underlying principle of SDM and SPM based face detection methods are similar. Instead of using two matrices to represent skin and non-skin probabilities, SDM uses a single matrix and updates its elements using winnows update rule for skin and non-skin

examples. Magnitude of the weight associated with each element of the sparse matrix known as SDM represents the likelihood of a corresponding pixel being skin or non-skin. A threshold is used to cutoff the weights below certain magnitude, which corresponds to non-skin pixels.

Table 2. TP and FP rates for SDM and SPM methods in Cb-Cr and T-S color spaces.

	Cb-Cr		T-S	
Skin Models	TP%	FP%	TP%	FP%
SDM	86.94	1.75	98.70	7.67
SPM	86.37	1.42	75.58	4.24

B. Face Color Constancy

We perform training on the face images from the PIE database [32] to learn the color changes on faces for different lighting conditions as described previously. We use only frontal faces with normal expression. The database has two sets of illumination variations. In set 1 the ambient lights are on and the point source is changing its state and position. In set 2 the ambient lights are off and again the point source is changing its state and position. We first use the set 1 to learn the joint color changes. Few images of a face taken under different illumination conditions from set 1 are shown in Fig. 14. Notice the changes in colors, shadows, and specular reflections for different illumination conditions.

We applied the FCC algorithm on 10 faces of arbitrary illumination conditions and transformed them to their reference illumination condition. The mean RMS error of the original faces with respect to the reference face was 70.51. We then computed the mean RMS errors of the transformed faces obtained using different number of eigenflows from the reference face (see Fig. 15, pg. 53).

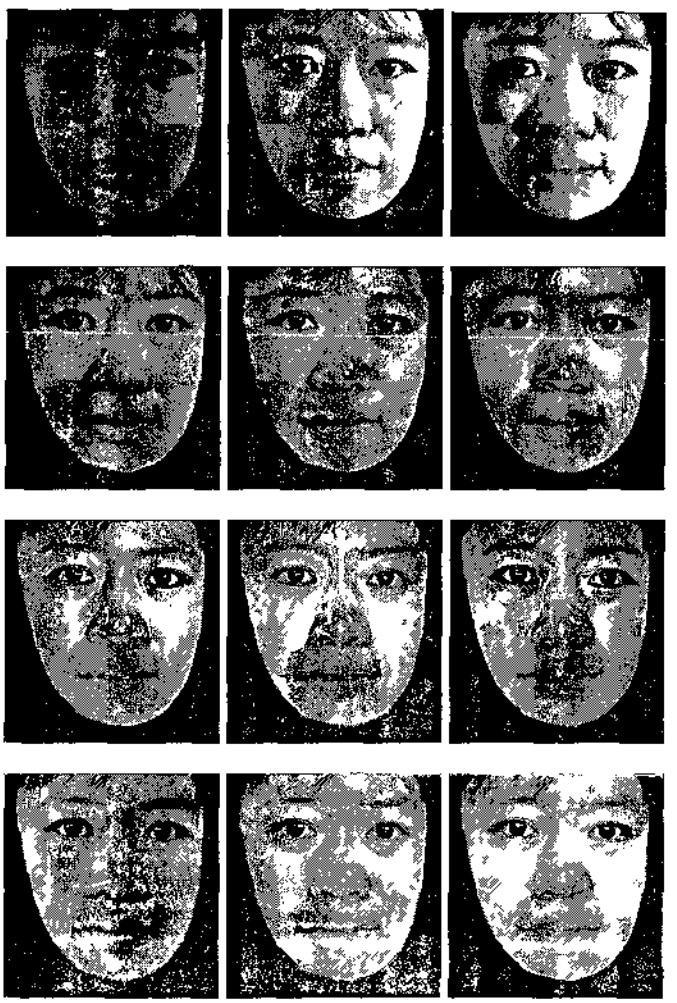


Fig. 14. Some of the face images of an individual from set 1 used for training.

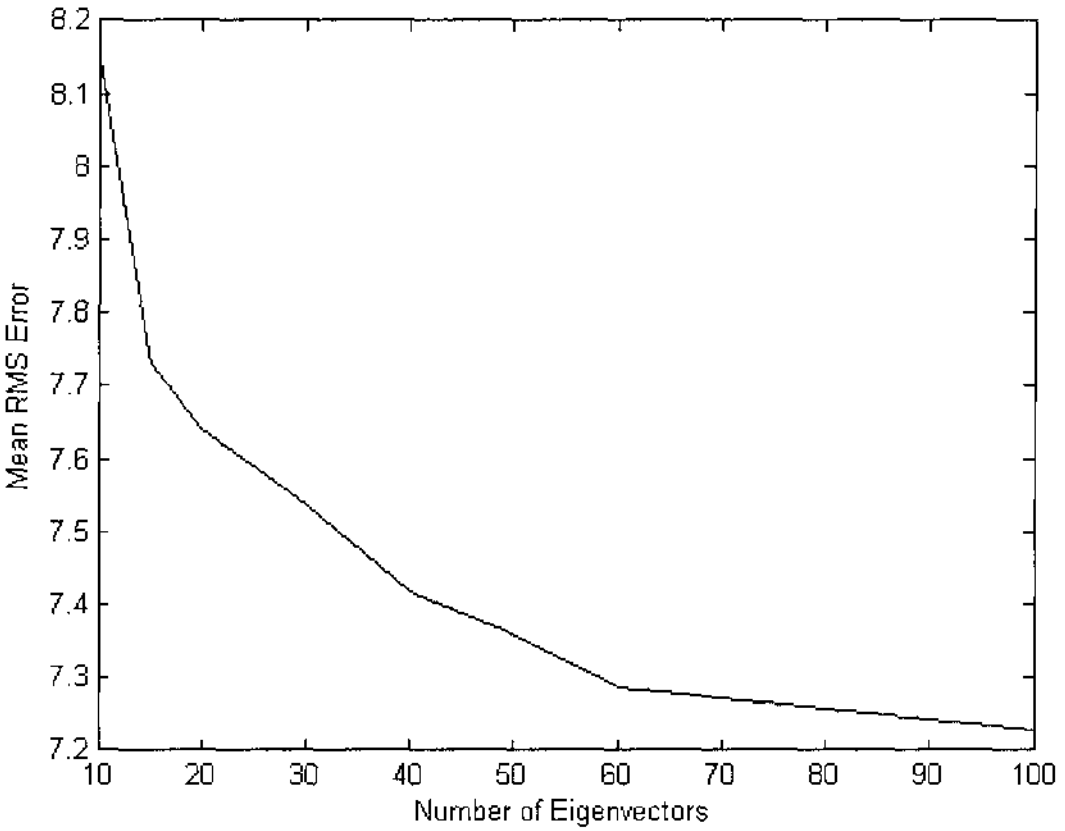


Fig. 15. Mean RMS error using different number of eigenflows. Mean RMS error of the original images was 70.51.

Using only 20 eigenflows the mean RMS error reduced to 7.65. Fig. 16 shows a few probe faces transformed towards the reference illumination condition. Notice that the effects of shadows and specular reflections are reduced and the color of the face is consistent with respect to the reference face. These results show that we were able to minimize the RMS error and transform the faces to reference illumination condition successfully. Thus, from the eigenflows obtained during the training procedure we compute the basis difference vectors for a probe face and efficiently transform it towards reference illumination condition of the gallery face. The individual shown in Fig. 16 was

not used during training to compute the eigenflows. This shows that our FCC algorithm is capable of generalizing color constancy for unseen faces. The results of FCC for unseen faces of different ethnicities are shown in Fig. 17.

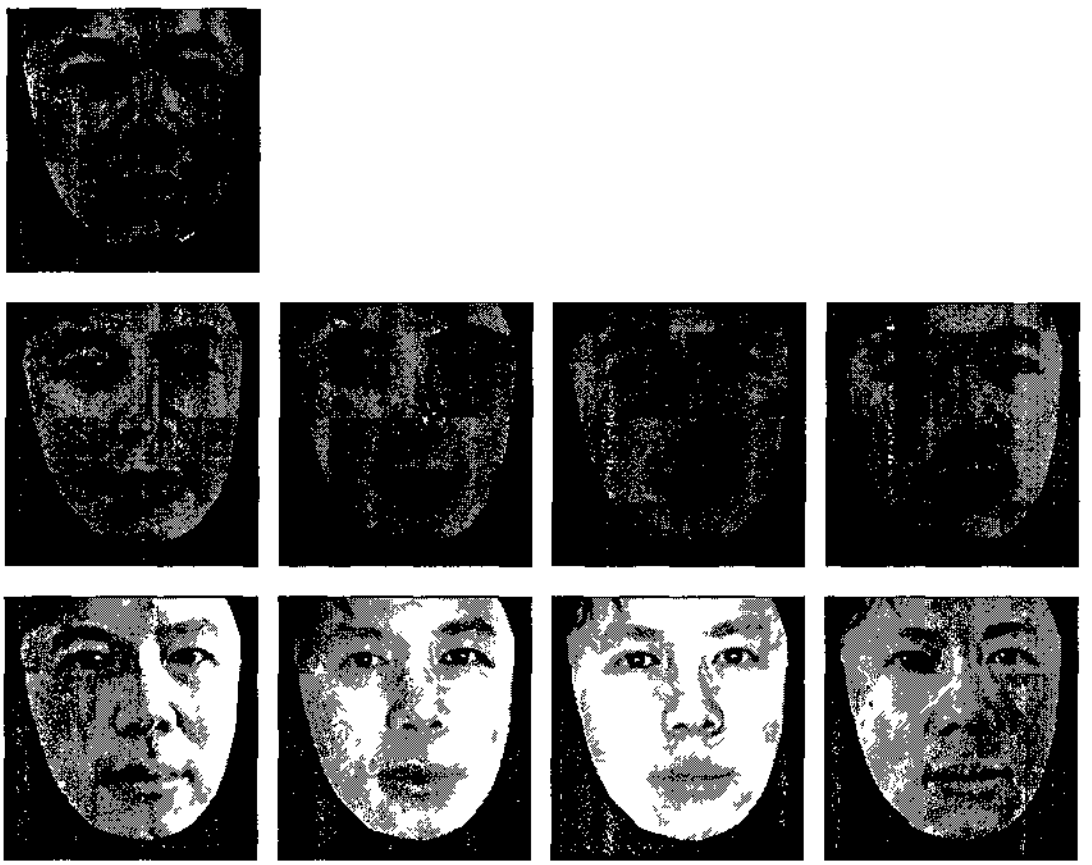


Fig 16. Probe faces (third row) are transformed (second row) to match the illumination condition of the reference face (first row).

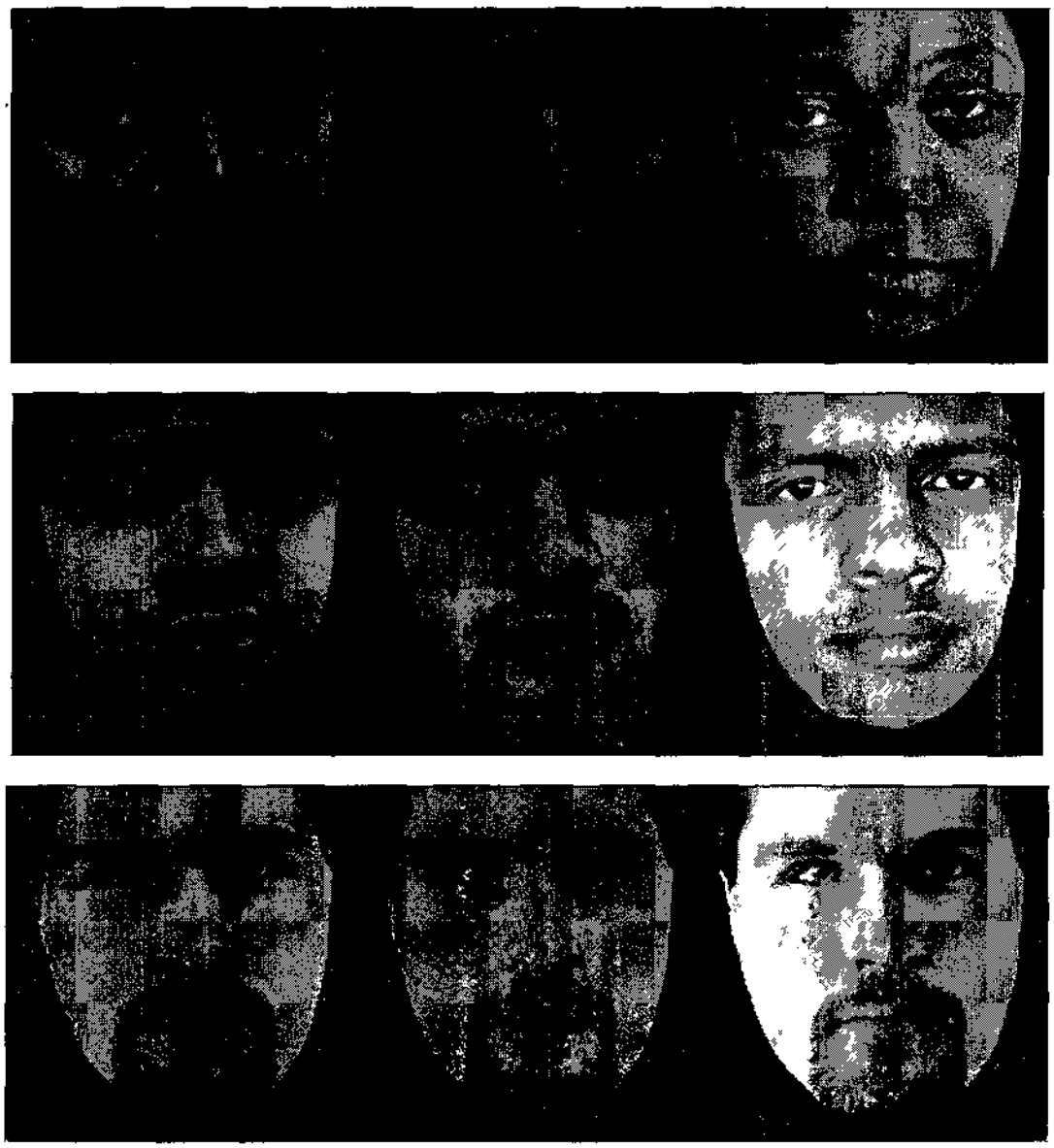


Fig. 17. Faces in the left column are references, middle column is the transformed faces and right column is the probe faces

C. Modular Face Recognition

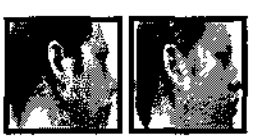
The performance of the conventional PCA based algorithm and the modular PCA based algorithm were evaluated with two image databases, Sheffield and Yale. The Sheffield database consists of images with varying pose and the Yale database consists of

images with varying illumination and expressions. All the images in both the databases were normalized and cropped to a size of 64×64 pixels.

1) Pose Invariance

For our tests we took a partial set of face images consisting of 10 images each of 20 different individuals from the Sheffield face database. Each image of a person is taken at a different pose, with a normal expression. Out of the ten images of a person, only eight were used for training and the remaining two were used to test the recognition rates. Fig. 18 shows the set of images of a person used for training and testing respectively. The choice of the training and testing images was made to test both the algorithms with head pose angles that lie outside the head pose angles they were trained with. The PCA and modular PCA methods may perform poorly with this selection of training and testing images, but our aim is to compare their performance for test images whose head pose lie outside the head pose of the training images.

(a)



(b)

Fig. 18. (a) Images of an individual used for training, (b) Images of an individual used for testing.

Fig. 19 shows the recognition rate, false recognition rate and false rejection rate for the modular PCA method with varying *N*. In the case of PCA the recognition rate was 0.3, false recognition rate was 0.625 and false rejection rate was 0.075.

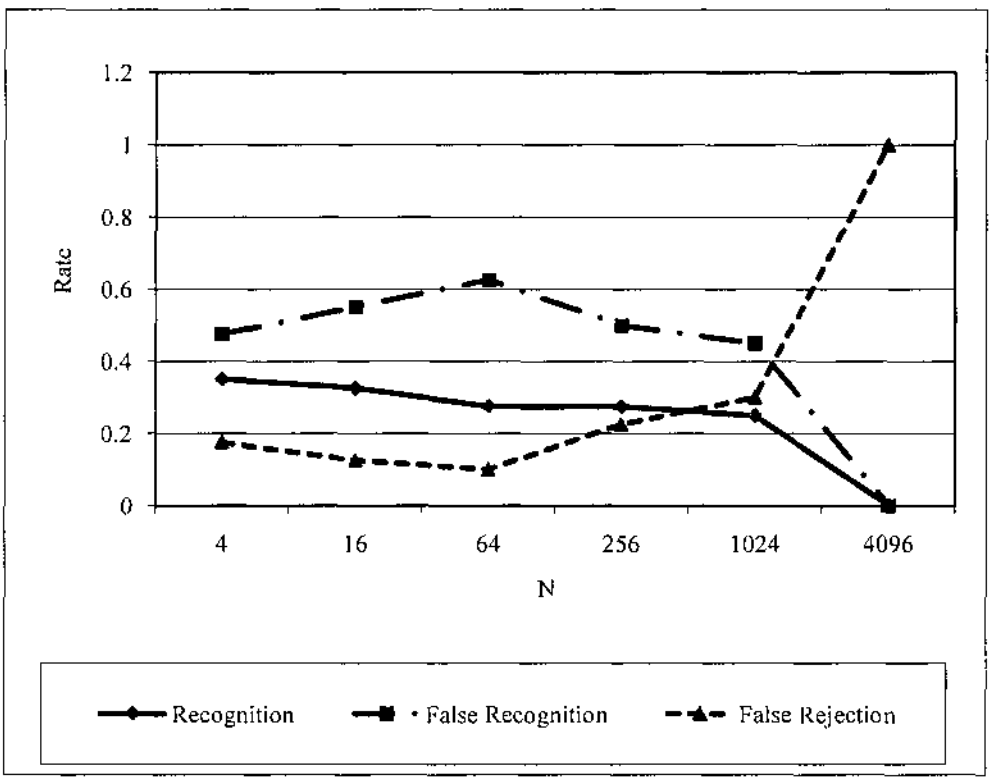


Fig. 19. Recognition, false recognition and false rejection rates of the modular PCA method for varying *N*. For PCA the rates were 0.3, 0.625 and 0.075 respectively.

From the results we note that the modular PCA method has a slightly better recognition rate and false recognition rate at N=4 and N=16, but the conventional PCA method has a slightly lesser false rejection rate. Hence the proposed method has no significant improvement over the PCA method under the condition of varying pose.

2) Illumination and Expression Invariance

The Yale database has 165 images of 15 adults, 11 images per person. The face images vary with respect to facial expression and illumination. The images have normal, sad, happy, sleepy, surprised, and winking expressions. There are also images where the position of the light source is at the center, left and right. In addition to these there are images with and without glasses. Out of the eleven images of a person, only eight were used for training and the remaining three were used to test the recognition rates. Fig. 20 shows the set of images of a person used for training and testing respectively. The choice of the training and test images was made to facilitate comparison of performance of both the methods for test images with uneven illumination and partial occlusion.

(a)

(b)

Fig. 20. (a) Images of an individual used for training, (b) Images of an individual used for testing.

We also conducted experiments by leaving out one image from each individuals set of 11 images during training and testing the recognition with the images left out. This was repeated 11 times by leaving out a different image each time. This kind of testing is known to as leave out one testing.

As before we vary the value of N from 4 to 4096 to observe the effect it has on face recognition. Fig. 21 shows the recognition rate, false recognition rate and false rejection rate for the modular PCA method with varying N. In the case of PCA the recognition rate was 0.44, false recognition rate was 0.31 and false rejection rate was 0.24. A second set of experiments were performed by leaving out one testing. The results obtained for modular PCA are shown in Fig. 22. For PCA, recognition rate was 0.48, false recognition rate was 0.36 and false rejection rate was 0.16.

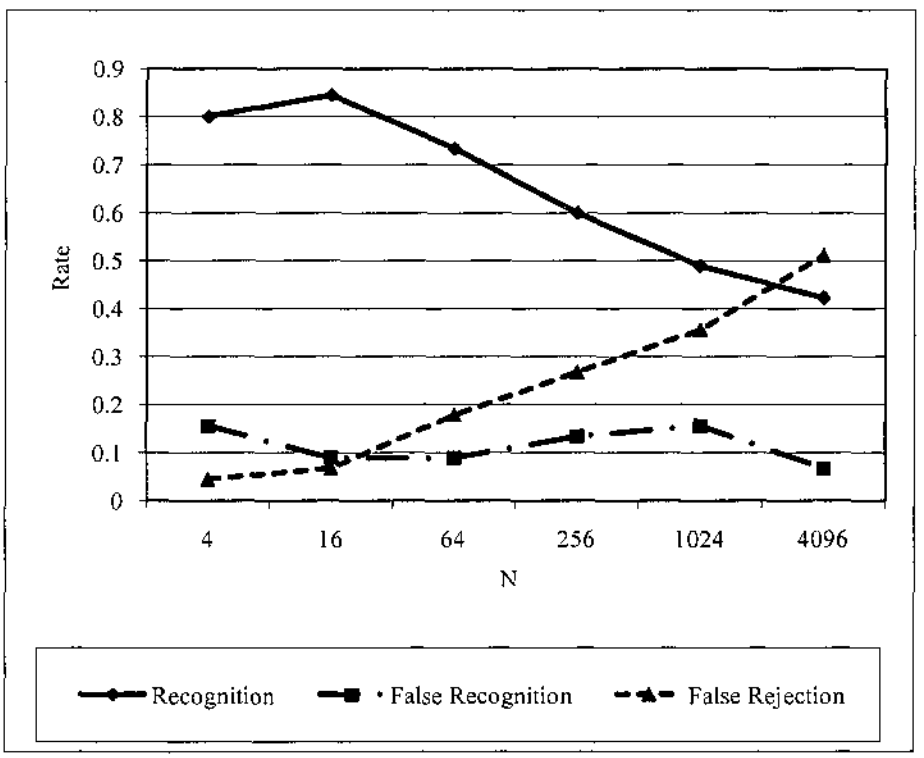


Fig. 21. Recognition, false recognition and false rejection rates of modular PCA method for varying N. For PCA rates were 0.44, 0.31 and 0.24 respectively.

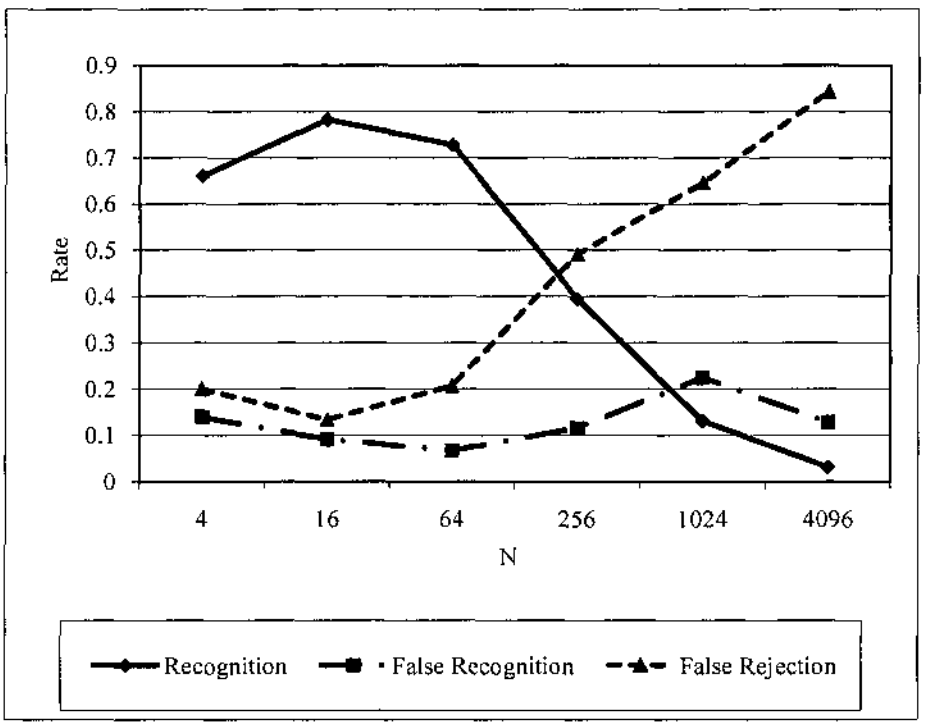


Fig. 22. Recognition, false recognition and false rejection rates of modular PCA method for varying *N*. For PCA rates were 0.48, 0.36 and 0.16 respectively.

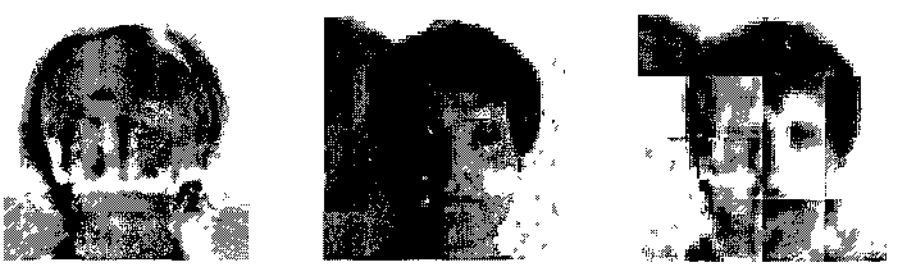


Fig. 23. Reconstructed images for a test image with varying illumination using the PCA and modular PCA method.

We observe from the results that the modular PCA method completely outperforms the PCA method in all aspects for *N* at 4, 16 and 64. However, best results were obtained for *N* at 16. Reconstruction of one of the test images was performed using PCA and

modular PCA for N at 16. The results of the reconstruction are shown in Fig. 23, the first image is the reconstructed image obtained using PCA method, the second image is the original image and the third image is the concatenation of the reconstructed images obtained using the modular PCA method for N=16.

D. Boosted Tree of Classifiers

In this section we present the results of face recognition and face verification experiments. We used the AR face database [31] for our experiments. This database has 26 frontal images captured in 2 sessions with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf) of 130 subjects. Each image is of size 768 x 576 pixels. For our experiments all images were converted to grayscale, cropped and aligned by the centers of eyes and mouth and resize to resolution of 16x16 pixels. A

Fig. 24. Images of a subject from AR face database.

mask is applied to remove the hairstyle and background information. Images from session one of a subject after pre-processing are shown in Fig. 24.

1) Unseen extra-personal variations

For the first experiment we trained AdaBoost and boosted tree with randomly selected 15 male and 15 female subjects. We generated the intra-personal differences from the all the 13 images from first session. To generate the extra-personal differences we used the 1st, 5th, 9th and 13th image from session 1 of each subject. We trained AdaBoost for $T = 50$ and $T = 100$ rounds. We also trained the boosted tree with $T = 50$. We tested the trained classifiers with a different set of 30 randomly selected subjects with equal number of males and females. We generated the intra-personal and extra-personal test data similar to training data.

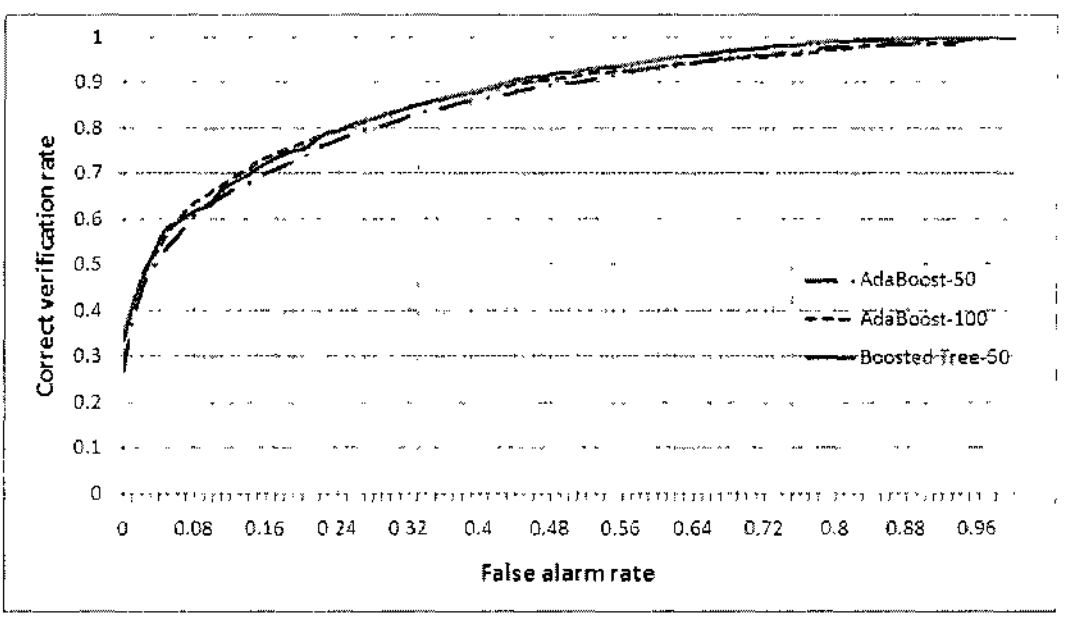


Fig. 25. ROC curves for AdaBoost-50, AdaBoost-100, and BoostedTree-50.

For face verification we computed the correct verification rate and false alarm rate for 2340 intra-personal samples and 6960 extra-personal test samples. We show the ROC

curves obtained for AdaBoost with 50 rounds of boosting (AdaBoost-50), AdaBoost with 100 rounds of boosting (AdaBoost-100) and boosted tree with 50 rounds of boosting (BoostedTree-50) in Fig. 25. The boosted tree was set to run for a maximum number of $L = 2$ levels.

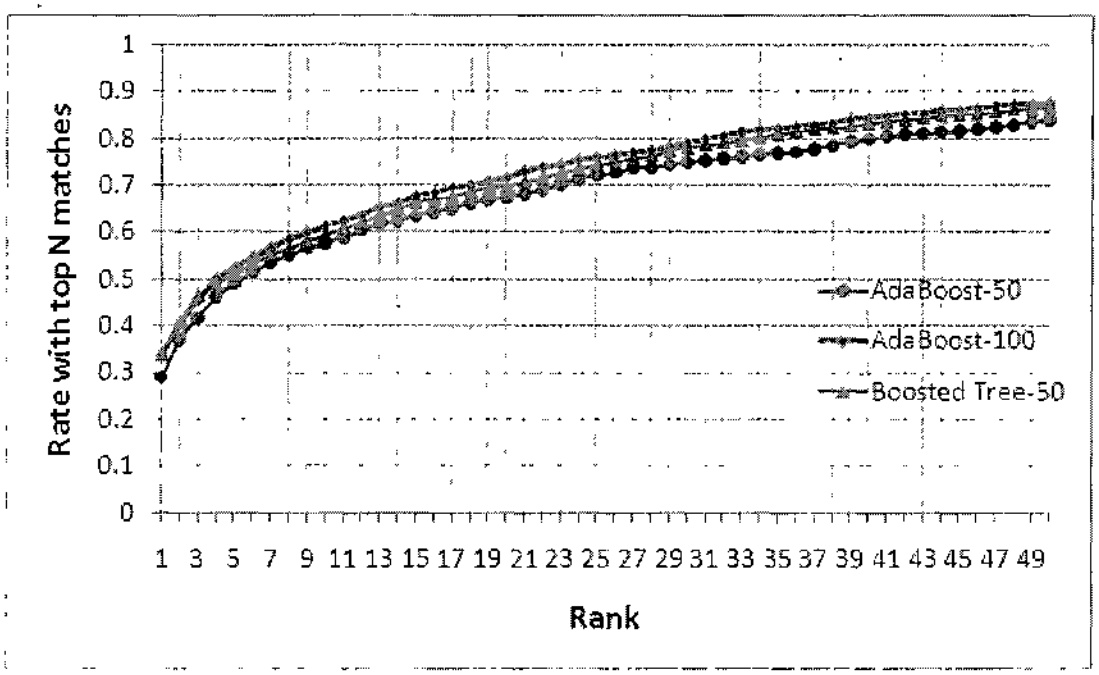


Fig. 26. Rank-N recognition rates for AdaBoost-50, AdaBoost-100, and BoostedTree-50.

For face recognition we only keep one image of the subject in the gallery, so for a probe image we will have 1 intra-personal pair and 236 extra-personal pairs. We do this with each of the 13 images for a subject. We then computed the rank-n recognition rate for AdaBoost-50, AdaBoost-100 and BoostedTree-50 as shown in Fig. 26.

In this experiment the subjects in training and test sets were different; however the intra-personal variations were same. Hence the classifiers were tested with extra-personal

variations which were not seen during training. We see from the results that the boosted tree method performs better than the AdaBoost-50 in this case.

2) Unseen intra-personal variations

In the next experiment we randomly select 30 male and 30 female subjects. For training we randomly select 7 images from session 1 and generate intra-personal data. Extra-personal data is generated from 2 images randomly selected from session 1 for all the 60 subjects. For testing we again randomly select 7 images from session 2 to generate intra-personal data. Extra-personal data is generated from 2 images randomly selected from session 2.

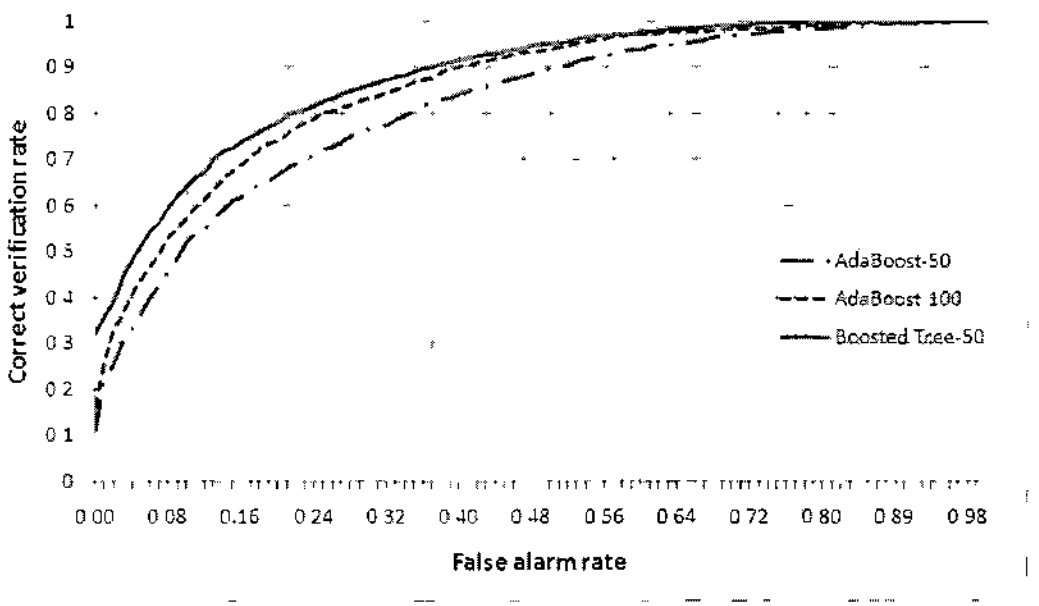


Fig. 27 ROC curves for AdaBoost-50, AdaBoost-100, and BoostedTree-50.

For face verification we computed the correct verification rate and false alarm rate for 1260 intra-personal samples and 7080 extra-personal test samples. We show the ROC

curves obtained for AdaBoost-50, AdaBoost-100 and BoostedTree-50 in Fig. 27. The boosted tree was set to run for a maximum number of $L = 10$ levels.

Again for the face recognition experiment we keep one image of the subject in the gallery, so for a probe image we will have 1 intra-personal pair and 118 extra-personal pairs. We do this with each of the 7 images for a subject. We then computed the rank-n recognition rate for AdaBoost-50, AdaBoost-100 and BoostedTree-50 as shown in Fig. 28.

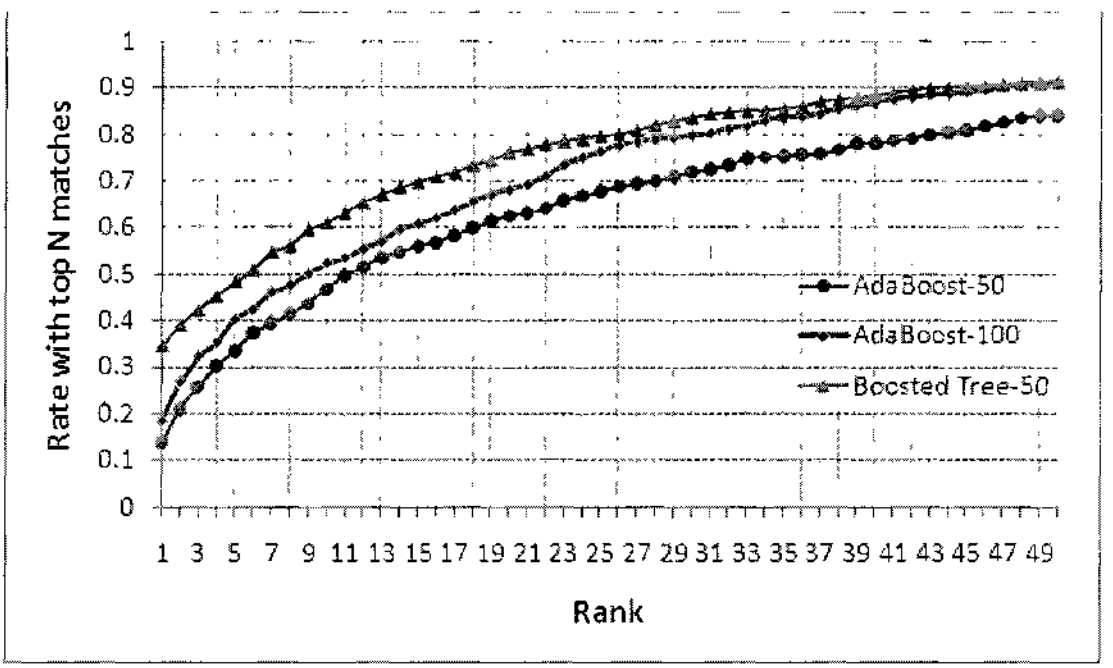


Fig. 28. Rank-N recognition rates for AdaBoost-50, AdaBoost-100, and BoostedTree-50.

In the second experiment the training and test images were randomly selected from different sessions for the same subjects. So the classifiers will be tested with intra-personal variations which were not seen during training. We see from the results that the

boosted tree method performs better than the AdaBoost-50 and AdaBoost-100 in this case as well.

VI. CONCLUSION

In this dissertation proposal we presented methods to improve the accuracy of face recognition systems. We made the following contributions:

- We proposed a new method called SDM (Skin Distribution Matrix) to perform skin segmentation using SNoW learning approach. The performance of skin segmentation using SDM and existing methods was evaluated on the Cb-Cr and T-S color spaces. We have shown that using SDM needs lesser resources than existing methods. The effectiveness of SDM was demonstrated by showing skin detection of images from different sources, which were not included in the training set.
- We developed a novel method to compensate of lighting changes on face images called face color constancy (FCC). The results show that FCC method is robust and can compensate for large illumination variations, shadows and specular reflections over faces. We also showed that our approach could have application in illumination invariant face recognition.
- We improved upon an existing method to do face recognition. Our method called modular PCA performs better than the PCA method under the conditions of large variations in expression and illumination. For large variations in pose there is slight improvement in the performance using modular PCA. This modular approach can be applied to any subspace algorithm to obtain the improvements we see in modular PCA approach.
- We demonstrated the use of boosted tree of classifiers in face verification and recognition. The proposed approach achieves better accuracy by learning a tree of

strong classifiers. This in effect naturally learns the clusters present in the intra-personal and extra-personal feature space. The use of probabilistic boosting tree frame work for face recognition is novel. The experimental results show that our approach is capable of performing better than a single boosted classifier. Even a single strong classifier trained for more rounds with AdaBoost is less accurate than our method trained with fewer rounds of boosting. We also showed that our method performs well for unseen intra-personal and extra-personal variations.

BIBLIOGRAPHY

[1] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[2] A. HyvÄrinen et al., *Independent Component Analysis*. John Wiley & Sons, 2001.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, October 1999.

[4] R. O. Duda et al., *Pattern Classification*. John Wiley & Sons, 2000.

[5] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[6] Y. Freund and R. E. Shapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. of Computer and System Sciences*, vol. 55, pp. 119-139, 1997.

[7] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. on Computer Vision*, 1999, pp. 1150-1157.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, pp. 91-110, 2004.

[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 257-263.

[10] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," School of Computer Science, Carnegie Mellon University and Intel Research, Tech. Rep. IRP-TR-03-15, Pittsburgh, PA, 2003.

[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[12] T. Ojala et al., "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[13] V. Takala et al., "Block-based methods for image retrieval using local binary patterns," in *Proc. Scandinavian Conf. on Image Analysis*, 2005, pp. 882-891.

[14] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Sciences*, vol. 6, no .2, pp. 559-572, 1901.

[15] H. Hotelling, "Analysis of a complex of statistical variables with principal components," *Journal of Educational Psychology*, vol. 24, pp. 417-441, 1933.

[16] R. Duda et al., *Pattern Classification*, Wiley-interscience Publication, 2001.

[17] E. Miller and K. Tieu, "Color Eigenflows: Statistical Modeling of Joint Color Changes," in *Proc. Int. Conf. on Computer Vision*, 2001, vol. 1, pp. 607-614.

[18] M.J. Jones and J.M. Regh, "Statistical color models with applications to skin detection," in *Proc. of Computer Vision and Pattern Recognition*, 1999, vol. 1, pp. 274-280.

[19] J. Brand and J.S. Mason, "A comparative assessment of three approaches to pixel-level human skin-detection," in *Proc. of Int. Conf. on Pattern Recognition*, 2000, vol. 1, pp. 1056-1059.

[20] M.-H. Yang and N. Ahuja, "Gaussian mixture model for human skin color and its application in image and video databases," in *Proc. of the SPIE: Storage and Retrieval for Image and Video Databases VII*, 1999, vol. 3656, pp. 458-466.

[21] D. Brown et al., "A SOM based approach to skin detection with application in real time systems," in *Proc. of the British Machine Vision Conference*, 2001.

[22] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," in *Proc. Automatic Face and Gesture Recognition*, 1996, pp. 236-241.

[23] M.-H. Yang et al., "A SNoW-Based Face Detector," *Advances in Neural Information Processing Systems*, vol. 12, pp. 855-861, 2000.

[24] H. Wu et al., "Face Detection From Color Images Using a Fuzzy Pattern Matching Method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 557-563, 1999.

[25] V. Vezhnevets et al., "A Survey on Pixel-Based Skin Color Detection Techniques," in *Proc. Graphicon*, 2003, pp. 85-92.

[26] A. Albiol et al., "Optimum color spaces for skin detection," in *Proc. of the Int. Conf. on Image Processing*, 2001, vol. 1, pp. 122-124.

[27] J.C. Terrillon and S. Akamasu, "Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images," in *Proc. of the 12th Conf. on Vision Interface*, 1999, vol. 2, pp. 180-187.

[28] G. Gomez, "On selecting color components for skin detection," in *Proc. of the Int. Conf. on Pattern Recognition*, 2002, vol. 2, pp. 961-964.

[29] D.B. Graham and N.M. Allinson, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition," H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie and T. S. Huang (eds), *Face Recognition: From Theory to Applications*, *NATO ASI Series F, Computer and Systems Sciences*, vol. 163, pp 446-456, 1998.

[30] P. N. Bellhumer et al., "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Face Recognition*, vol. 17, no. 7, pp. 711-720, 1997.

[31] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Tech. Rep. #24, June 1998.

[32] T. Sim et al., "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[33] B. Moghaddam et al., "Beyond Eigenfaces: Probabilistic Matching for Face Recognition," in *Proceedings. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 1998, pp. 30-35.

[34] M. J. Jones, and P. Viola, "Face Recognition Using Boosted Local Features," MERL Tech. Rep., TR 2003-25, April 2003.

[35] P. Yang et al., "Face Recognition Using Ada-Boosted Gabor Features," in *Proc. 6th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2004, pp 356-361.

[36] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," in *Proc. 10th IEEE Int. Conf. on Computer Vision*, 2005, vol. 2, pp. 1589-1596.

[37] B. Wu and R. Nevatia, "Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection," in *Proc. of IEEE 11th Int. Conf. on Computer Vision*, 2007, pp. 1-8.

[38] L. Zhang et al., "Boosting Local Feature Based Classifiers for Face Recognition," in *Proc. Of IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Face Processing in Video*, 2004, pp 87-87.

[39] C. Huang et al., "Vector boosting for rotation invariant multi-view face detection," in *Proc. of 10th IEEE Int. Conf. on Computer Vision*, 2005, vol. 1, pp. 446-453.

[40] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Journal of Machine Learning*, vol. 37, no. 3, pp. 297--336, 1999.

VITA

Rajkiran Gottumukkal is a doctoral student in Electrical and Computer Engineering at Old Dominion University. Rajkiran received the Bachelor's degree in Electronics & Communication Engineering from Bangalore University, India, in 1999. He received Master's degree in Electrical Engineering from Old Dominion University, in 2003.

Rajkiran Gottumukkal is an experienced researcher and developer of computer vision, image processing and machine learning software. He is currently working as Software Engineer at Charles River Laboratories. In this position he develops software to segment and detect objects of interest in tissue images. This software is being used to guide pathologists in analyzing tissue images more accurately. Prior to that Rajkiran worked from 2006 to 2009 as a computer vision engineer at Behavioral Recognition Systems. Here he was involved in research and development of video analysis software AISight™. This software is being used to automatically detect abnormal activities in surveillance videos.