CHAPTER-1

INTRODUCTION

1.1. OVERVIEW

In this report we proposes a face recognition system for consumer applications under that is invariant to the illumination variation, frontal face with varying expression, occlusion and disguise. The proposed method based on face recognition system to recognize the face under the above problems of the vast verities. For this purpose we have recognize the face using the sparse representation with the feature extraction. Although sparse representation classifier can recognize without feature extraction but feature extraction may make the face recognition system more efficient. Objective of this report is to improve the accuracy of face recognition under real life environments. Since differential components between lines rarely vary in relation to illumination direction, we expect the proposed feature to be able to cope with illumination variation. Moreover, the advantage of our approach is that it does not require any complex pre-processing steps like other methods, and it can be easily implemented in a real-time face recognition system since the proposed feature is simply obtained from LBP and Gabor collectively. In the proposed method, the face images are first copied into two images one image feed to the LBP feature extraction and other one feed to Gabor feature extraction, and both methods are then separately applied to each image. Next, from each image features are computed and then concatenated to make feature vector. This feature vector is then used for recognizing the face using sparse representation classifier. Performance evaluation of the proposed system was carried out using an extended Yale face database B which consists of 2,414 face images for 38 subjects representing 64 illumination conditions under the frontal pose. Consequently, we will demonstrate the effectiveness of the proposed approach by comparing our experimental results to those obtained with conventional approaches.

1.2. COMMON PROBLEMS WITH FACE RECOGNITION SYSTEM

Most general problems face recognition system can face are given below

1.2.1. Illumination variation

Illumination variation are the conditions in which the uncontrolled illumination is applied to the face or object of the interest. Illumination variation usually cause the performance degradation to most of the face recognition system. Example of the shown in picture 1[1]

Picture 1: The picture shows the illumination variation[2], in the above image illumination variation occurs due to spotlight is put on different angles due to these angles shadow forms on one side of the face.

1.2.2. Alignment problem

The alignment problem is that the face is not aligned to the origin of the image due to that the face is not recognized by many system. Shown in picture 2

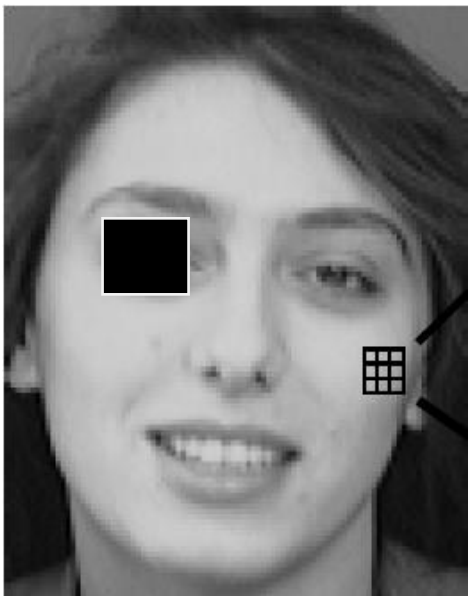
Picture 2(a) Picture 2(b)

Picture 2: The above image show misalignment of the face in the image picture 2(a) shows original image picture 2(b) shows misaligned image

1.2.3. Occlusion

Occlusion occurs when some object comes in front of the face to be recognized. This problem is a very critical because only the few systems are developed for the occlusion

handling and may not work properly for the problem so we try to eliminate that problem. Example shown in figure below



Picture 3: shows the condition of the occlusion

1.2.4. Disguise

This is the condition occurs when the occlusion to the face occurs because of the facial accessories for example hat, sunglasses, scarf, mask. This happens in day to day life for example you take a picture of man after a week braid and mustaches grown over the face many system may not recognize the face. We will try to overcome this problem to some extent

Figure 4(a)

Figure 4(b)

Picture 4: Condition of disguise figure 4, figure 4 (a) shows person with hat figure 4(b) person with sunglasses.

1.2.5. Age variation

Age variation is the condition for which we have trained the system for the conditions of the age variation. With humans age variation condition always associated with the change in time of the observation. With increase in time face of the human changes as baby, childhood, teenage, young, mature and old etc. these classification are according to the group of ages. Although, this algorithm would not work in the condition of age variation but I shows this problem just to make aware the face system user to the same.

Picture 5(a) picture 5(b)

Picture 5; 5(a) shows childhood image and 5(b) shows image of young person

1.2.6. Very Low Resolution Problem

This condition occurs when the subject to be recognized is very far from the camera in that case the face to be recognized picture snap shoot become very difficult. Our algorithm also would not work for this condition.

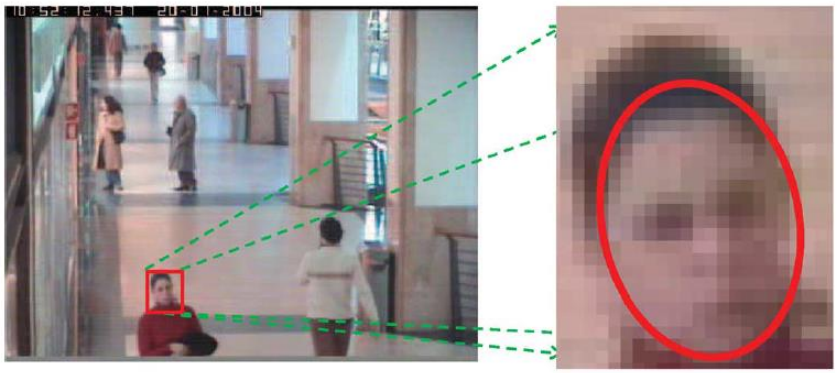


Figure 6 (b)shows the photo in which person is far from camera 6(b) shows low resolution face.

1.3. ORGANIZATION OF THESIS

Chapter 2: This chapter includes the Algorithm of the proposed Model. It gives overview of the overall system.

Chapter 3: This chapter is about the Linear Binary Pattern. This chapter tells about the LBP feature extraction method.

Chapter 4: This chapter is about the Gabor Feature Extraction. This chapter also gives the mathematical formulation of Gabor Filter.

Chapter 5: This chapter includes the mathematical formulation of the Sparse Representation classifier and tells how it can be used with or without feature extraction schemes.

Chapter 6: This chapter discusses the Experiment and results. In experiment is carried out on two different databases and shows respective results.

Chapter 7: In this chapter conclusion and future scope is discussed.

CHAPTER-2

ALGORITHM OF THE PROPOSED SYSTEM

In this model we try to develop a face recognition system which can work with alignment and occlusion problem. For this we use the feature extraction along with the sparse representation classifier

Algorithm of the proposed model

2.1. Training algorithm

Input: path of the training images

Output: Dictionary of the feature vectors for the training images

Step 1: path of the training image

Step 2: loop according to the training images required

Step 3: preprocessing

Step 4: LBP feature extraction

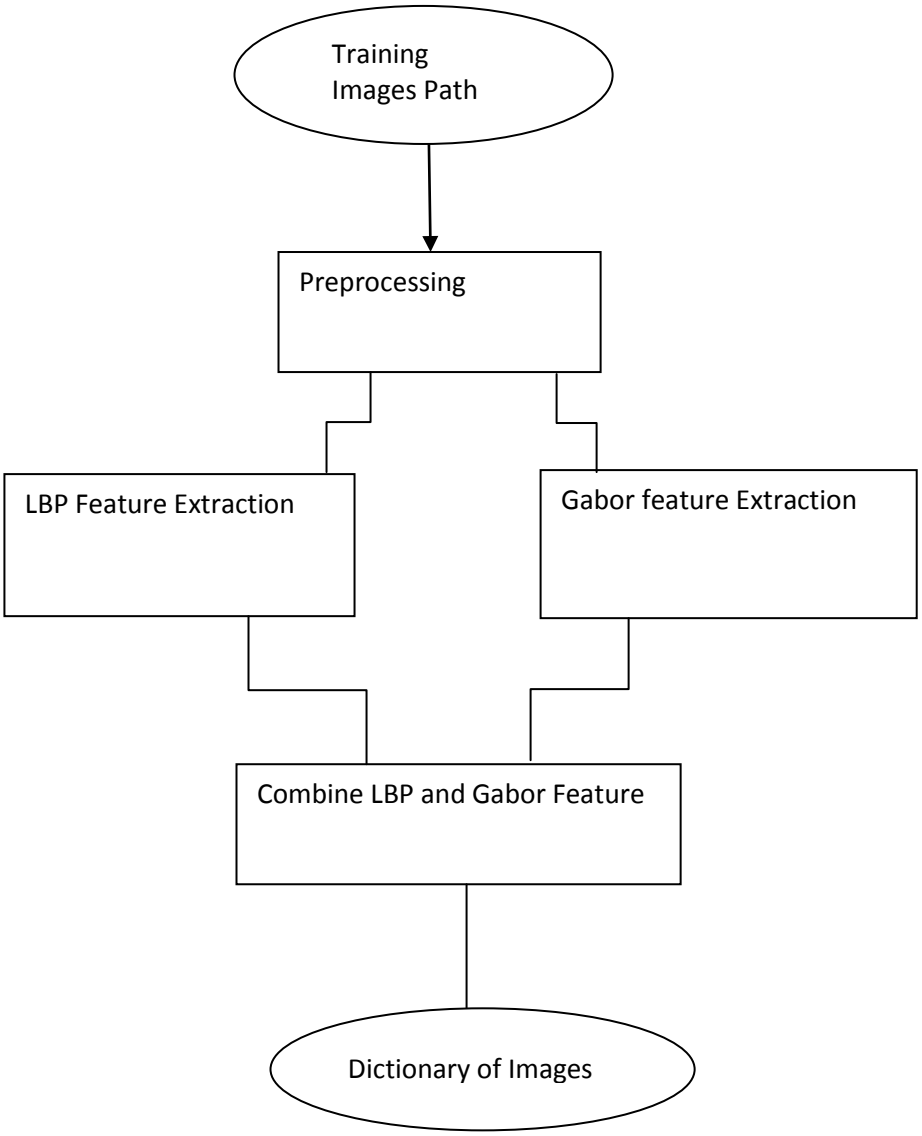
Step 5: Gabor feature extraction

Step 6: Combine the LBP and Gabor Feature to make descriptor

Step 7: Create Dictionary

Step 8: repeat for N times

Training algorithm in case of the proposed system only the feature of the both types LBP and Gabor are calculated, and then stored in a form suitable for the storage in the storage media. The features are extracted from all the training examples and make feature



Picture 1: Proposed Training System Block Diagram

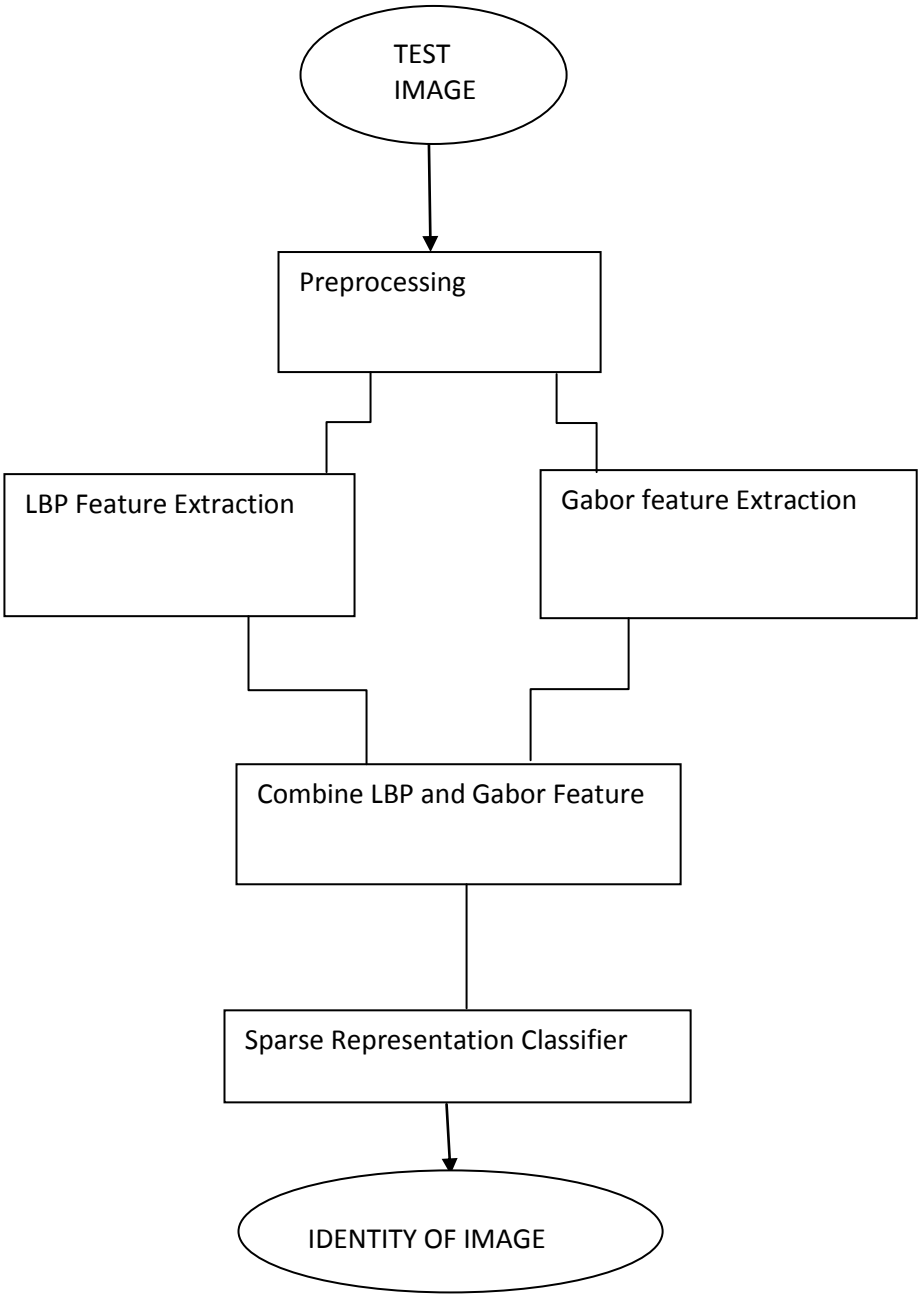
2.2. Testing algorithm

Input: Test Image

Output: Identity of the image

Step 1: Preprocessing

Step 2: Apply LBP for feature extraction



Picture 2: Proposed Testing System Block Diagram

Step 3: Apply Gabor filter for feature extraction

Step 4: Combine the LBP and Gabor Feature to make descriptor

Step 5: Compare testing image feature with Dictionary of feature vectors

The above image shows the block diagram of the proposed system in which the input is the image of the subject to be identified and the output is the identity of the image it shows the name and identified image from the database.

Preprocessing Step

Preprocessing may be the cropping the face image of the concerned subject which may be cropped manually or if you want to crop it automatically from given image you may introduce the face detection algorithm before the recognition algorithm to make it fully automatically. But here, we are doing only recognition and not detection. So, the preprocessing face image must be cut manually.

In many algorithm the image must be converted from RGB to Gray image this is because color feature are the worst feature due to the color varies in accordance to the lightning conditions to remove the impact of the color variation to the processing algorithm we must eliminate the color component from the image for the RGB to Gray conversion is the best tool in the matlab to do best with the problem occurs due to the color component of the image.

Now we have to parallel system which are used for the feature extraction purpose for doing this we have two feature extraction system to improve efficiency of the recognition system.

1) LBP Feature extraction

2) Gabor Feature extraction

For extraction of the feature the one copy of the image to be recognized feed to the LBP feature Extraction system and other one feed to the Gabor Feature extraction system. The extracted feature are thus concatenated to make single feature vector.

Creating Dictionary during Training System

This feature vector is then stored to make dictionary for the system for the comparison at the time of the testing. The features for all the training images are calculated at this time so the more time is required at this stage. We may also compare test image by extracting feature for training images but it require large time and greatly reduce the performance so we calculate feature vector for the training image once and make the dictionary. So the dictionary creation is the most critical issue in this the images to be selected for making the dictionary may be the issue of thinking because which images to be chosen decides the efficiency for the recognition system. We choose the set of images for creating dictionary such that the system may give the maximum performance for this remove the redundant images for example images with the same

pose and expression are the worst case for creating the dictionary due to the redundant feature vectors are stored for these image the result remains same but the recognition reduced greatly because of the unnecessarily matching the test image with the same training images.

Testing system

The feature vector for the testing image is calculated exactly same as that of the training due to the size of the feature vector of the testing image must be same as that of the training image, if they are not the message of the mismatch in the dimension of the feature occur and the system would not work. So, for the proper working of the system dimension of the training feature vector and the testing feature vector must be same. Hence we use the fixed length descriptor.

Another thing a testing system require is classifier. A classifier is a tool used for comparing the feature of the training image and the testing image the efficiency and the working of the system greatly depends upon the selection of the classifier after the feature extraction technique. We use the sparse representation classifier for recognizing the person for this face recognition algorithm. The features extracted from both LBP and Gabor are then feed to sparse representation classifier, it then compare the features by using the sparse measure and ℓ^1 –minimization technique.

Above one is the overview of system for more about the system and its components read following chapters.

CHAPTER-3

LOCAL BINARY PATTERNS FEATURE EXTRACTION

Real world textures can occur at arbitrary rotations and they may be subjected to varying illumination conditions. This has inspired few studies on gray scale and rotation invariant texture analysis, which presented methods for incorporating both types of invariance. Gray scale invariance by assuming that the gray scale transformation is a linear function.

In this study we propose a theoretically and computationally simple approach which is robust in terms of gray scale variations and which is shown to discriminaterotated textures efficiently. Extending our earlier work, we present a truly gray scale and rotation invariant texture operator based on local binary patterns[18]. Starting from the joint distribution of gray values of a circularly symmetric neighbor set of eight pixels in a 3x3 neighborhood, we derive an operator that is by definition invariant against any monotonic transformation of the gray scale. Rotation invariance is achieved by recognizing that this gray scale invariant operator incorporates a fixed set of rotation invariant patterns[19].

The novel contribution of this work is to use only a limited subset of 'uniform' patterns instead of all rotation invariant patterns, which improves the rotation invariance considerably. We call this operator LBP_8^{riu2}. The use of only 'uniform' patterns is motivated by the reasoning that they tolerate rotation better because they contain fewer spatial transitions exposed to unwanted changes upon rotation. This approximation is also supported by the fact that these 'uniform' patterns tend to dominate in deterministic textures, which is demonstrated using a sample image data. Further, we propose operator called LBP_{16}^{riu2}, which enhances the angular resolution of LBP_8^{riu2} by considering a circularly symmetric set of 16 pixels in a 5x5 neighborhood[20].

These operators are excellent measures of the spatial structure of local image texture, but they by definition discard the other important property of local image texture, contrast, since it depends on the gray scale. We characterize contrast with rotation invariant variance measures named VAR_8 and VAR_{16}, corresponding to the circularly symmetric neighbor set where they are computed. We present the joint distributions of these complementary measures as powerful tools for rotation invariant texture classification. As the classification rule we employ nonparametric discrimination of sample and prototype distributions based on a log-likelihood measure of the (dis)similarity of histograms[21].

The performance of the proposed approach is demonstrated with two problems used in recent studies on rotation invariant texture classification. In addition to the original experimental setups we also consider more challenging cases, where the texture classifier is trained at one particular rotation angle and then tested with samples from other rotation angles. Excellent experimental results demonstrate that the texture representation obtained at a specific rotation angle generalizes to other rotation angles. The proposed operators are also computationally attractive, as they can be realized with a few operations in a small neighborhood and a lookup table.

3.1 MATHEMTICAL OVERVIEW OF LBP

Gray Scale and Rotation Invariant Local Binary Patterns

We start the derivation of our gray scale and rotation invariant texture operator by

defining texture *T* in a local 3x3 neighborhood of a monochrome texture image as the joint distribution of the gray levels of the nine image pixels:

T= *P*(g0,g1,g2,g3,g4, g5, g6, g*7,g8)*

Where g_t (i=0,...,8), correspond to the gray values of the pixels in the 3x3 neighborhood according to the spatial layout illustrated in Fig. 1. The gray values of diagonal pixels (g_2, g_4, g_6, and g_8) are determined by interpolation.

g4 •	*g3* •	*g2* •
g5	og*0*	**g1*
• *g6*	• *g7*	**g8*

Fig. 1. The circularly symmetric neighbor set of eight pixels in a 3x3 neighborhood.

Achieving Gray Scale Invariance

As the first step towards gray scale invariance we subtract, without losing information, the gray value of the center pixel (g_0) from the gray values of the eight surrounding pixels of the circularly symmetric neighborhood (g, i=l,...,8) giving:

T = p(go, g**l**-g**o**, g**2** – g**o**, g**3** – g**o**, g**4** – g**o**, g**5** – g**o**, g**6** – g**o**, g**7** – g**o**, *g8* - g**o**) (2)

Next, we assume that differences $g_r g_0$ are independent of g_0, which allows us to factorize Eq.(2):

T ≈ *p(*g**o**)p(g**l** – g**o**, g**2** – g**o**, g**3** – g**o**, g**4** – g**o**, g**5** – g**o**, g**6** – g**o**, g**7** – g**o**, g**8** - g**o**) (3)

In practice an exact independence is not warranted, hence the factorized distribution is only an approximation of the joint distribution. However, we are willing to accept the possible small loss in information, as it allows us to achieve invariance with respect to shifts in gray scale. Namely, the distribution $p(g_0)$ in Eq.(3) describes the overall luminance of the image, which is unrelated to local image texture, and consequently does not provide useful information for texture analysis. Hence, much of the information in the original joint gray level distribution (Eq.(l)) about the textural characteristics is conveyed by the joint difference distribution:

T≈ p(gi-go, g2- go,g3-go, g4-go, g5-go, g6-go, g7-go, g8-go) (4)

Signed differences $g_r g_0$ are not affected by changes in mean luminance, hence the joint difference distribution is invariant against gray scale shifts. We achieve invariance with respect to the scaling of the gray scale by considering just the signs of the differences instead of their exact values[19]:

T ≈ *p(s(*g**l** – g**o**), *s(g**2** — g**o**), *s(g**3** – g**o**), *s(*g**4** – g**o**), ...,s(g**8**- g**o**)) (5)

Where

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (6)$$

If we formulate Eq.(5) slightly differently, we obtain an expression similar to the LBP (Local Binary Pattern) operator we proposed in

$$LBP_8 = \sum_{i=1}^{8} (g_i - g_0) 2^{i-1} \qquad (7)$$

The two differences between LBP_8 and the LBP operator the pixels in the neighbor set are indexed so that they form a circular chain, and 2) the gray values of the diagonal pixels are determined by interpolation. Both modifications are necessary to obtain the circularly symmetric neighbor set, which allows for deriving a rotation invariant version of LBP_8. For notational reasons we augment LBP with subscript 8 to denote that the LBP_8 operator is determined from the 8 pixels in a 3x3 neighborhood. The name 'Local Binary Pattern' reflects the nature of the operator, i.e. a local neighborhood is thresholded at the gray value of the center pixel into a binary pattern[19]. LBP_8 operator is by definition invariant against any monotonic transformation of the gray scale, i.e. Achieving Rotation Invariance

The LBP_8 operator produces 256 (2^8) different output values, corresponding to the 256 different binary patterns that can be formed by the eight pixels in the neighbor set. When the image is rotated, the gray values g_i will correspondingly move along the perimeter of the circle around g_0. Since we always assign g_1 to be the gray value of element (0,1), to the right of g_0, rotating a particular binary pattern naturally results in a different LBP_8 value. This does not apply to patterns 00000000_2 and $11nnn_2$ which remain constant at all rotation angles. To remove the effect of rotation, i.e. to assign a unique identifier to each rotation invariant local binary pattern we define:

$$LBP_8^{ri36} = \min\{ROR(LBP_8, i) where, i = 0.1.....7\} \qquad (8)$$

as long as the order of the gray values stays the same, the output of the LBP_8 operator remains constant.

where *ROR(x,i)* performs a circular bit-wise right shift on the 8-bit number *xi* times. In terms of image pixels Eq.(8) simply corresponds to rotating the neighbor set clockwise so many times that a maximal number of the most significant bits, starting from g_8, are 0. We observe that LBP_8^{ri36} can have 36 different values, corresponding to the 36 unique rotation invariant local binary patterns illustrated in Fig. 2, hence the super- script^{ri36}. LBP_8^{ri36} quantifies the occurrence statistics of these patterns corresponding to certain microfeatures in the image, hence the patterns can be considered as feature detectors. For example, pattern #0 detects bright spots, #8 dark spots and flat areas, and #4 edges. Hence, we have obtained the gray scale and rotation invariant operator LBP_8^{ri36} that we designated as LBPROT

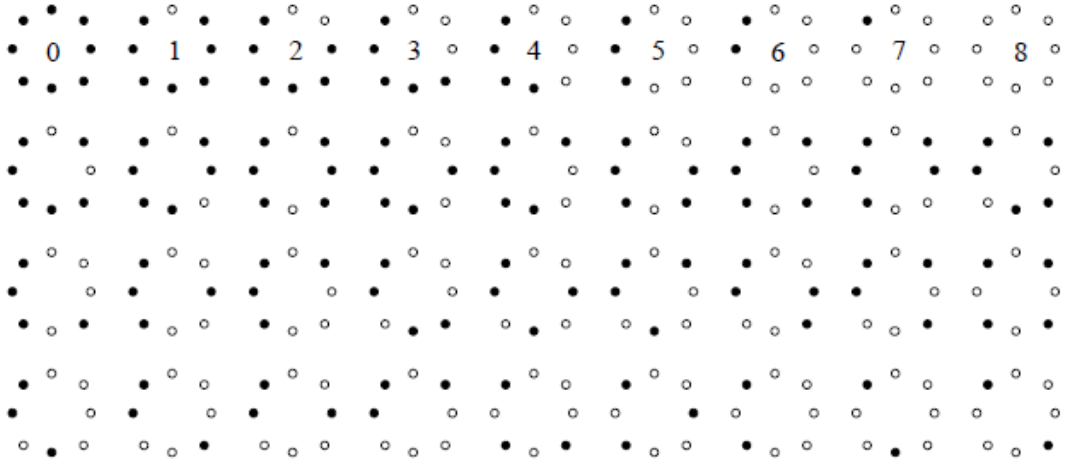


Fig. 2. The 36 unique rotation invariant binary patterns that can occur in the eight pixel circularly symmetric neighbor set. Black and white circles correspond to bit values of 0 and 1 in the 8-bit output of the LBP8 operator. The first row contains the nine 'uniform' patterns, and the numbers inside them correspond to their unique LBP_8^{riu2} values.

Improved Rotation Invariance with 'Uniform' Patterns

However, experiments has showed that LBP_8^{ri36} as such does not provide a very good discrimination, as we also concluded. There are two reasons:

1. The performance of the 36 individual patterns in discrimination of rotated textures varies greatly: while some patterns sustain rotation quite well, other patterns do not and only confuse the analysis. Consequently, using all 36 patterns leads to a suboptimal result (addressed in this section).
2 .Crude quantization of the angular space at 45^o intervals .

The varying performance of individual patterns attributes to the spatial structure of the patterns. To quantify this we define an uniformity measure U('pattern'), which corresponds to the number of spatial transitions (bitwise 0/1 changes) in the 'pattern'. For example, patterns 00000000_2 and 11111111_2 have U value of 0, while the other seven patterns in the first row of Fig. 2 have U value of 2, as there are exactly two 0/1 transitions in the pattern. Similarly, other 27 patterns have U value of at least 4.

We argue that the larger the uniformity value U of a pattern is, i.e. the larger number of spatial transitions occurs in the pattern, the more likely the pattern is to change to a different pattern upon rotation in digital domain. Based on this argument we designate patterns that have U value of at most 2 as 'uniform' and propose the following

$$LBP_8^{riu2} = \begin{cases} \sum_{i=1}^{8} s(g_i - g_0), & \text{if } U(LBP_8) \le 2 \\ 9 & \text{otherwise} \end{cases} \qquad (9)$$

Eq.(9) corresponds to giving an unique label to the nine 'uniform' patterns illustrated in the first row of Fig. 2 (label corresponds to the number of '1' bits in the pattern), the 27 other patterns being grouped under the 'miscellaneous' label (9). Superscript ^{riu2} corresponds to the use of rotation invariant 'uniform' patterns that have U value of at most 2.

The choice of 'uniform' patterns with the simultaneous compression of 'non-uniform' patterns is also supported by the fact that the former tend to dominate in deterministic

textures. Using the image data of the experiments. In practice the mapping from LBP_8 to LBP_8^{riu2}, which has 10 distinct output values, is best carry out with a lookup table of 256 elements.

Improved Angular Resolution with a 16 Pixel Neighborhood

The rotation invariance of LBP_8^{riu2} is hampered by the crude 45^o quantization of the angular space provided by the neighbor set of eight pixels. To report this we present a modification, where the angular space is quantized at a finer resolution of 22.5^o intervals. This is attained through the circularly symmetric neighbor set of 16 pixels illustrated. Yet again, the gray values of neighbors which do not fall exactly in the center of pixels are estimated by interpolation. Note that we increase the size of the local neighborhood to 5x5 pixels, as the eight added neighbors would not provide too much new information if inserted into the 3x3 neighborhood. An added benefit is the dissimilar spatial resolution, if we should want to perform multi-resolution analysis.

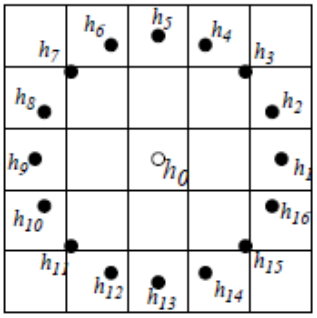


Fig. 3. The circularly symmetric neighbor set of 16 pixels in a 5x5 neighborhood.

Following the derivation of LBP8, we first define the 16-bit version of the rotation variant LBP:

$$LBP_{16} = \sum_{i=1}^{16} s(h_i - h_0)2^{i-1} \qquad (10)$$

The LBP_{16} operator has 65536 (2^{16}) different output values and 243 different rotation invariant patterns can occur in the circularly symmetric set of 16 pixels. Choosing again the 'uniform' rotation invariant patterns that have at most two 0/1 transitions, we define LBP^{riu2},

$$LBP_{16}^{riu2} = \begin{cases} \sum_{i=1}^{16} s(h_i - h_0)^2, & \text{if } U(LBP_{16}) \le 2 \\ 17 & \text{otherwise} \end{cases} \qquad (11)$$

Therefore, the LBP_{16}^{riu2} operator takes 18 distinct output values, of which values from 0 (pattern 0000000000000000_2) to 16 (pattern 1111111111111111_2) correspond to the number of 1 bits in the 17 unique 'uniform' rotation invariant patterns, and value 17 denotes the 'miscellaneous' class of all 'nonuniform' patterns. Actually the mapping from LBP_{16} to LBP_{16}^{riu2} is implemented with a lookup table of 2^{16} elements.

Rotation Invariant Variance Measures of the Contrast of Local Image Texture

Normally, image texture is considered as a two dimensional phenomenon that be able to characterized with two orthogonal properties, spatial structure (pattern) and contrast (the 'amount' of local image texture). With gray scale and rotation invariant texture description these two are an interesting pair: whereas spatial pattern is affected by

rotation, contrast is not, and *vice versa*, while contrast is affected by the gray scale, spatial pattern is not. Subsequently, till we want to restrict ourselves to pure gray scale invariant texture analysis, contrast is of no interest, as it depends on the gray scale.

The LBP_8^{riu2} and LBP_{16}^{riu2} operators are true gray scale invariant measures, i.e. their output is not affected by any monotonic transformation of the gray scale. They are excellent measures of the spatial pattern, but by definition throw away contrast. If we under stable lighting conditions wanted to incorporate the contrast of local image texture as well, we can measure it with rotation invariant measures of local variance:

$$VAR_8 = \frac{1}{8}\sum_{i=1}^{8}(g_i - \mu_8)^2, \text{ where } \mu_8 = \frac{1}{8}\sum_{i=1}^{8} g_i \qquad (12)$$

$$VAR_{16} = \frac{1}{16}\sum_{i=1}^{16}(h_i - \mu_{16})^2, \text{ where } \mu_{16} = \frac{1}{16}\sum_{i=1}^{16} h_i \qquad (13)$$

VAR_8 and VAR_{16} are by definition invariant against shifts in gray scale. Meanwhile LBP and VAR are complementary, their joint distributions LBP_8^{riu2}/VAR_8 and LBP_{16}^{riu2}/VAR_{16} are very powerful rotation invariant measures of local image texture.

Nonparametric Classification Principle

In the classification phase a test sample *S* was allocated to the class of the model *M* that maximized the log-likelihood measure:

$$L(S, M) = \sum_{b=1}^{B} S_b \log M_b \qquad (14)$$

where *B* is the number of bins, and S_b and M_b correspond to the sample and model probabilities at bin *b*, respectively. This nonparametric (pseudo-)metric measures likelihoods that samples are from alternative texture classes, based on exact probabilities of feature values of pre-classified texture prototypes. In the case of the joint distributions LBP_8^{riu2}/VAR_8 and LBP_{16}^{riu2}/VAR_{16}, the log-likelihood measure (Eq.(14)) was extended in a direct manner to scan through the two-dimensional histograms.

Sample and model distributions were achieved by scanning the texture samples and prototypes with the selected operator, and dividing the distributions of operator outputs into histograms having a fixed number of *B* bins. Since LBP_8^{riu2} and LBP_{16}^{riu2} have a entirely defined set of discrete output values, they do not need any additional binning procedure, but the operator outputs are directly gathered into a histogram of 10 (LBP_8^{riu2}) or 18 (LBP_{16}^{riu2}) bins.

Variance find out VAR_8 and VAR_{16} have a continuous-valued output, hence quantization of their feature space is requisite. This had been done by adding together feature distributions for every single model image in a total distribution, which was divided into *B* bins having an equal number of entries. From now, the cut values of the bins of the histograms related to the (100/B) percentile of the joint data. Deriving the

cut values from the total distribution and assigning every bin the same amount of the combined data assures that the highest resolution of quantization is used where the amount of entries is largest and *vice versa.* The number of bins used in the quantization of the feature space is significant, as histograms with a too modest number of bins fail to deliver sufficient discriminative information about the distributions. Oppositely, since the distributions have a finite number of entries, a too large number of bins may lead to sparse and unstable histograms. As a rule of thumb, statistics literature often proposes that an average number of 10 entries per bin should be sufficient. In the experiments we set the value of *B* so that this condition was satisfied.

3.2. LOCAL BINARY PATTERN PIXEL BASED

Local binary patterns were introduced by Ojala et al as a fine scale texture descriptor. In its simplest form, an LBP description of a pixel is created by thresholding the values of the 3 x 3 neighborhood of the pixel against the central pixel and interpreting the result as a binary number. The process is illustrated in figure 1.

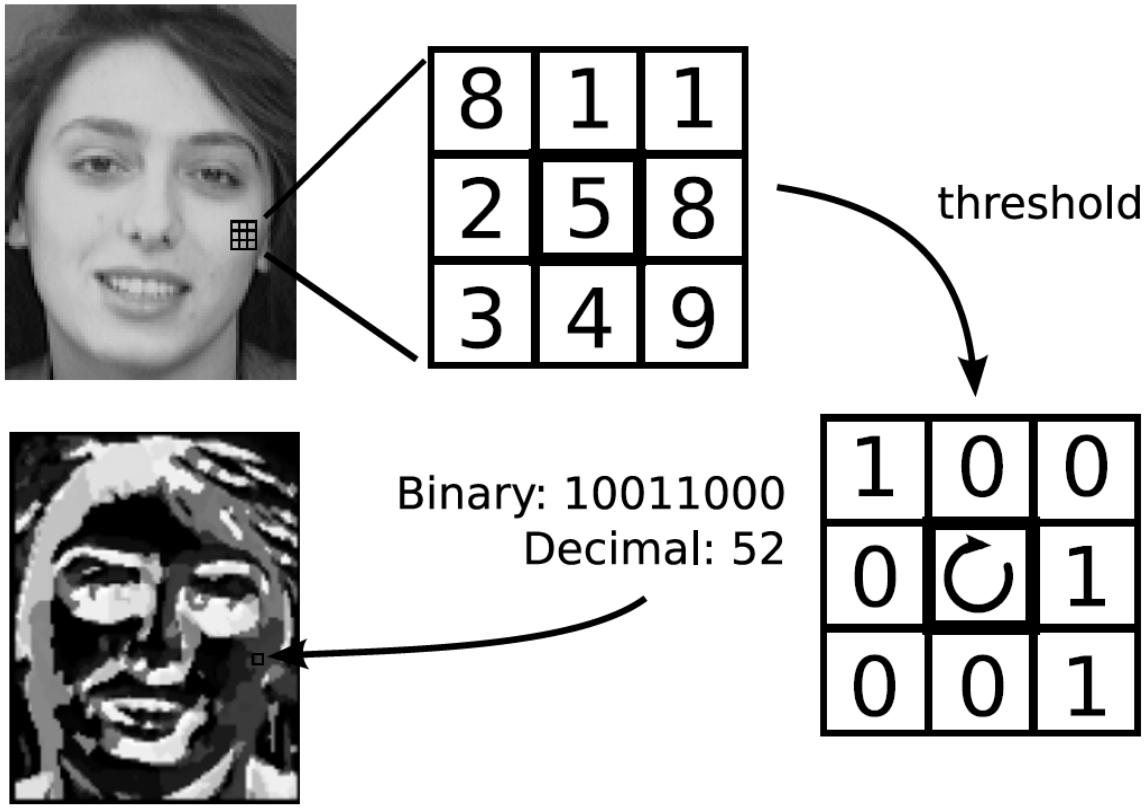


Figure 1. The LBP operator thresholds each pixel against its neighboring pixels and interprets the result as a binary number. In the bottom image each gray-level value corresponds to a different local binary pattern.

In the LBP operator is generalized by allowing larger neighborhood radii r and different number of sampling points s. These parameters are indicated by the notation LBPs,r. For

example, the original LBP operator with radius of 1 pixel and 8 sampling points is $LBP_{8,1}$. Another

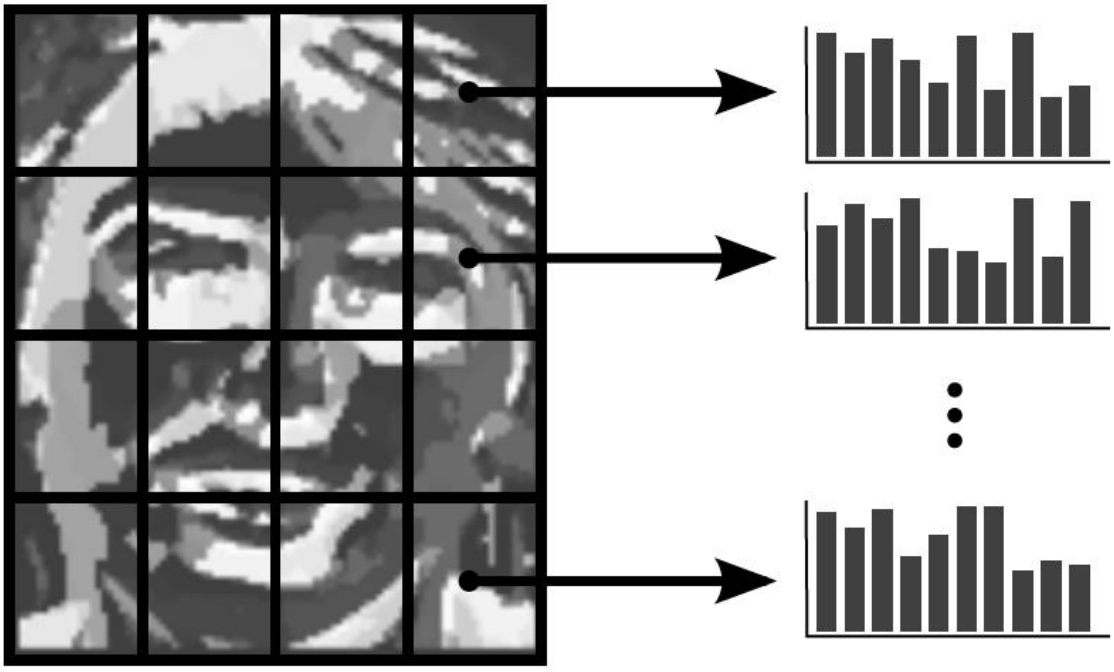


Figure 2. LBP descriptors are built by partitioning the LBP face image into a grid and computing LBP histograms over each grid cell. These histograms may then be concatenated into a vector or treated as individual descriptors.

important extension is the definition of "uniform patterns". An LBP[19] is defined as uniform if it contains at most two 0-1 or 1-0 transitions when viewed as a circular bit string. Thus the 8-bit strings 01100000 and 00000000 are uniform, while 01010000 and 00011010 are not. Ojala observed that when using 8 sampling points, uniform patterns accounted for nearly 90% of the patterns in their image datasets. Therefore, little information is lost by assigning all non uniform patterns to a single arbitrary number. Since only 58 of the 256 possible 8 bit patterns are uniform, this enables significant space savings when building LBP histograms. To indicate the usage of two-transition uniform patterns, the superscript u2 is added to the LBP operator notation. Hence the LBP operator with a 2 pixel radius, 8 sampling points and uniform patterns is known as LBP.

The success of LBP has inspired several variations. These include local ternary patterns, elongated local binary patterns, multi scale LBP [20], centralized binary patterns and patch based LBP, among others.

In this we use LBP, which was selected by Ahonen et al in their pioneering work applying LBP to face recognition. This descriptor has been used, by itself or in combination with other features, by most methods that use LBP[21] for face recognition.

3.2. HISTOGRAM

In our experiment we have applied the eight different filter of 3*3 matrix to extract the LBP features by multiplying separately to the face image. After applying the filter histogram is calculated of all the eight filtered images. These histogram then concatenated to form the feature vector.

For a grayscale image there are different 256 gray values. This means any of the pixel in grayscale image falls in one of 256 category. Histogram of an image is a graph which shows the number of pixels falls into the one of 0-255(or 1-256) gray value. It gives the no pixels against no gray values. It tells how many pixels haves same values every face image have different histogram after applying LBP so histogram of the LBP filtered image may be used as the feature vector.

Figure 3: x axis represents the 256 different gray values and y axis no of pixels having that value

CHAPTER-4

GABOR FEATURE EXTRACTION

4.1. GABOR FILTER

Gabor filter is a linear filter used for edge detection used for the edge detection, it is named after Dennis Gabor. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly suitable for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave[27].

Simple cells in the visual cortex of mammalian brains shall be modeled by Gabor functions. Thus, image analysis with Gabor filters is thought to be similar to perception in the human visual system. Gabor filter impulse response is defined by a sinusoidal wave and by a plane wave for 2D Gabor filters [7], multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. This filter have a real as well as an imaginary component representing both orthogonal directions. These components can be formed to make a complex number or used individually.

Complex equation of the Gabor Filter [1]

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

Real part

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Imaginary part

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Where the values of

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

In the above equation, λ represents the wavelength of the sinusoidal factor, θ characterizes the orientation of the normal to the parallel stripes of a Gabor function, ψ represents the phase offset, σ represents the sigma/standard deviation of the Gaussian envelope and γ symbolized the spatial aspect ratio, and identifies the ellipticity of the support of the Gabor function.

4.2. FEATURE EXTRACTION

A group of Gabor filters with distinct frequencies and orientations would be useful for extracting useful features from a face image. Gabor filters are widely used in pattern analysis applications.

Four orientations are shown on the right 0°, 45°, 90° and 135°. The original character picture and the superposition of all four orientations are shown on the left. Gabor filters are directly related to Gabor wavelets, due to they shall be designed for a number of dilations and rotations. However, generally expansion is not applied for Gabor wavelets, meanwhile this requires computation of bi-orthogonal wavelets, which may be very time-consuming. Hence, typically, a filter bank consisting of Gabor filters [4] with various scales and rotations is created. The filters are convolved with the signal, resulting in a so-called Gabor space. This process is closely related to processes in the primary visual cortex. Jones and Palmer showed that the real part of the complex Gabor function is a good fit to the receptive field weight functions found in simple cells in a cat's striate cortex. Histogram of Gabor output can be taken for feature extraction [29].

The Gabor space is very useful in image processing applications such as optical character recognition, iris recognition and fingerprint recognition. Relationships among activations for a specific spatial location are very distinguishing between objects in an image. Moreover, significant initiations can be pull out from the Gabor space with the purpose of create a sparse object representation.

4.3. GTP PATTERN DESCRIPTOR

The detected regions must be normalized to a fixed size, a local descriptor is constructed within each region as follows: We first apply the Gabor filter to each image patch. We use Gabor filters because they provide good perception of local image structures and they are robust to illumination variations. The Gabor kernels [4] are defined as

$$\psi_{\mu,\nu}(x,y) = \frac{\|\mathbf{k}_{\mu,\nu}\|^2}{\sigma^2}\exp\left(-\frac{\|\mathbf{k}_{\mu,\nu}\|^2\|\mathbf{z}\|^2}{2\sigma^2}\right) \times \left[\exp\left(i\mathbf{k}_{\mu,\nu}^T\mathbf{z}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right], \quad \text{(eq1)}$$

where μ and ν define the orientation and scale of the Gabor kernels, respectively, z = $(x, y)^T$, and the wave vector $k_{\mu,\nu}$ is defined as

$$\mathbf{k}_{\mu,\nu} = (k_\nu \cos\phi_\mu, k_\nu \sin\phi_\mu)^T, \quad \text{(eq2)}$$

with $k_\nu = k_{max}/f^\nu$, $k_{max} = \pi/2$, $f = \sqrt{2}$, and $\phi = \pi\mu/8$. Due to the relatively small region size (40 x 40 pixels) we process Gabor kernels at a single scale ($\nu = 0$) and four orientations ($\mu \in \{0,2,4,6\}$, corresponding to 0, 45, 90, and 135 degrees) with $\sigma = 1$ are used. Furthermore, we only use the odd Gabor kernels (imaginary part), which are sensitive to edges and their locations. These four Gabor kernels are able to discriminate local details in the face image. Four Gabor filtered images for a local

patch. These four response images emphasize edges in four different orientations (0, 45, 90, and 135 degrees).

For each pixel (x, y) in the normalized keypoint region, there are four Gabor filter responses as follows:

$$f_i(x,y) = G_i(x,y) * I(x,y), i = 0,1,2,3 \qquad \text{(eq3)}$$

where $G_i = imag(\psi_{\alpha,0})$ is the i th odd Gabor kernel and * is the convolution operator. The responses of the four filters are combined as a ternary pattern:

$$GTP_t(x,y) = \sum_{i=0}^{3} 3^i [(f_i(x,y) < -t) + 2(f_i(x,y) > t)], \qquad \text{(eq4)}$$

where t is a small positive threshold (a value of 0.03 is used in our experiments). We call this local descriptor the GTP[1]. It encodes local structures from the responses of odd Gabor filters in four different orientations. The local ternary pattern provides a discriminative encoding of the four Gabor filters

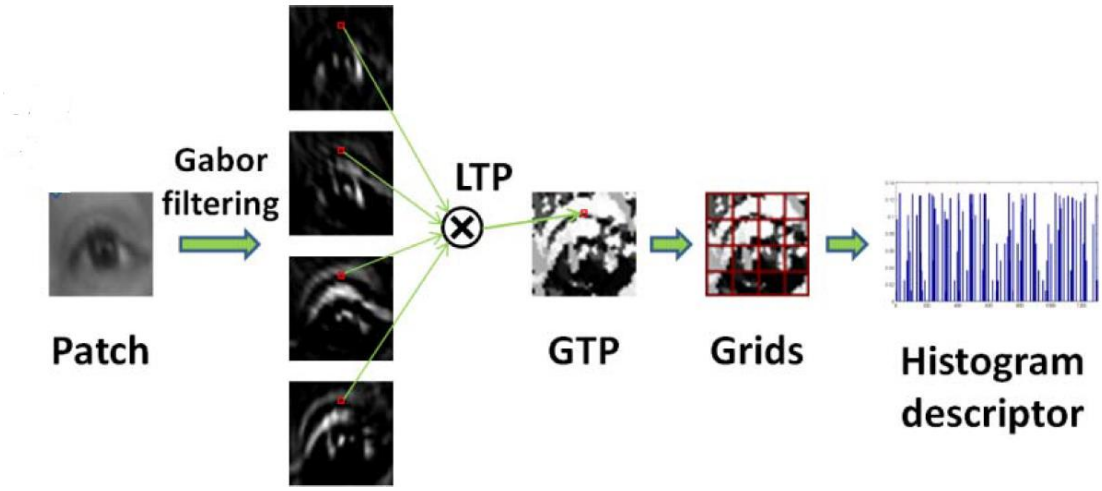


Figure shows Gabor Feature Extraction by creating GTP Pattern

CHAPTER-5

SPARSE REPRESENTATION

Automatically recognizing human faces from frontal views with varying expression and illumination, as well as occlusion and disguise is a complex problem. Design of the recognition problem as one of classifying among multiple linear regression models and claim this theory from sparse signal representation offers the strategy to addressing this problem. Based on a sparse representation computed by $\ell 1$-minimization [5], this introduces a common classification algorithm for face recognition. This framework offers new understandings into two critical problems in face recognition: feature extraction and robustness to occlusion. For feature extraction, we show that if sparsity in the recognition problem is properly coupled, the choice of features remains no longer critical. What is necessary is whether the number of features are sufficiently large and whether the sparse representation is appropriately calculated. Alternative features such as downsampled images and random projections works just as good as conventional features such as Eigenfaces and Laplacianfaces, if the dimension of the feature space exceeds certain threshold, predicted by the theory of sparse representation [4]. Then the framework can handle errors due to occlusion. The t sparse representation aids to forecast how much occlusion the recognition procedure be able to handle and how to pick the training images to make best use of robustness to occlusion.

We achieve the discriminative Characteristics of sparse representation to perform classification. Rather than using the nonspecific dictionaries, we represent the test sample in an overcomplete dictionary those base elements are the training samples themselves. If enough training samples are existing from each class, there is a possibility to denote the test samples as a linear mixture of just those training samples from the same class. This representation is naturally sparse, including only a minor portion of the overall training database. We debate that in various problems of interest, this is in fact the sparsest linear representation of the test sample in terms of this dictionary and can be recovered well via ℓ^1 -minimization. Looking for the sparsest representation hence automatically differentiates between the various classes existing in the training set. Sparse representation too provides an easy and unexpectedly effective way of rejecting invalid test samples not arising from any class in the training database. These samples' sparsest representations be subject to involve many dictionary elements, covering multiple classes.

In place of using sparsity to classify appropriate model or appropriate features that can far ahead be used for classifying all test samples, it uses the sparse representation of each individual test sample directly for classification, adaptively selecting the training samples those give the maximum compact representation. The suggested classifier will be considered a generalization of widely used classifiers such as nearest neighbor (NN) and nearest subspace (NS) (i.e., least distance to the subspace spanned all training samples from every object class). NN classifies the test sample based on the best representation in relations to a single training sample, however NS classifies based on the best linear representation in relations to all the training samples in each class. The nearest feature line (NFL) algorithm makes a balance among these two extremes, classifying based on the best affine representation is related to a pair of

training samples. This method strikes a similar balance but considers all possible supports (within each class or across multiple classes) and adaptively selects the minimal amount of training samples needed to represent each test sample.

Motivating and studying this approach for classification inside the context of automatic face recognition. Human faces are perhaps the most broadly studied object in image-based recognition. This is because of the notable face recognition capability of the human visual system and also because of a large amount of important applications for face recognition. Additionally, technical issues associated with face recognition are representative of object recognition and even data classification in general. On the other hand, the theory of sparse representation and compressed sensing lead to new visions into two crucial issues in automatic face recognition: the feature extraction and the occlusion.

Feature extraction

Low-dimensional features of an object image are the most appropriate or useful for classification is a dominant issue in face recognition and in object recognition in general. An huge volume of works has been devoted to investigate various data-dependent feature transformations for projecting the high-dimensional test image into lower dimensional feature spaces: For example include Eigenfaces, Laplacianfaces, Fisherfaces etc. Using so many offered features and so little consensus about which are better or worse, there is absence of procedures to decide which features to use. Though, in our suggested framework, the theory of compressed sensing implies that the exact choice of feature space is no longer precarious: Even random features hold enough information to recover the sparse representation and hence correctly classify any test image. What is critical is that the dimension of the feature space is sufficiently large and that the sparse representation is correctly computed.

Occlusion handling

Occlusion poses a major difficulty to robust real-world face recognition. This trouble is mainly because of the random nature of the error experienced by occlusion. This may affect any part of the image and may be arbitrarily enormous in magnitude. However, this error naturally corrupts only a fraction of the image pixels and is hence sparse in the usual basis given by individual pixels. While the error has such a sparse representation, it can be handled uniformly with this framework. The center in which the error is sparse can be treated as a distinct class of training samples. The succeeding sparse representation of an occluded test image pertaining to this prolonged dictionary (training images plus error basis) naturally split up the component of the test image arising due to occlusion from the component arising from the identity of the test subject. In this context, the concept of sparse representation and compressed sensing illustrates when such source-and- error separation can take place and consequently how much occlusion the resulting recognition algorithm can bear.

The proposed system is of extensively concern to object recognition in general, the studies and experimental results in this are confined to human frontal face recognition.

Working of Sparse Representation Classifier

An elementary difficulty in object recognition is to use labeled training samples from k different object classes to properly determine the class to which a new test sample belongs. We arrange the given ni training samples from the *i*th class as columns of a matrix $A= [v_{i,l}, v_{i,2},..., v_{i,ni}] \in IR^{mxni}$. In the context of face recognition, we will

identify a w x h grayscale image with the vector $v \in IR^m$ (m = wh) given by stacking its columns; the columns of Ai are then the training face images of the *i*th subject.
Test Sample as a Sparse Linear Combination of Training Samples

A vast variety of statistical, generative, or discriminative models have been offered for take advantage of the structure of the Ai for recognition. One particularly easy and effective style of models the samples from a single class as lying on a linear subspace. Subspace models are flexible sufficient to capture abundant of the dissimilarity in real data sets and are specifically well motivated from the perspective of face recognition, wherever it has been observed that the images of faces in varying lighting and expression lie on a special low-dimensional subspace, is called a face subspace. Even though the proposed framework and algorithm may also put on to multimodal or nonlinear distributions, for simplicity of presentation, we first assume that the training samples from a only class do lie on a subspace. This is the individual former knowledge about the training samples we will be using in our solution.

Specified essential training samples of the *i*th object class, $Ai = [v_{i,l}, v_{i,2},..., v_{i,ni}] \in IR^{mxn_i}$, some different (test) sample $y \in IR^m$ from the same class will approximately lie in the linear span of the training samples related with object *i*:

$$Y= \alpha_{i,l}, v_{i,l+} \alpha_{i,2,} v_{i,2+\ldots.+} \alpha_{i,ni,} v_{i,ni} \quad \text{(eq1)}$$

for some scalars, $\alpha_{i,j} \in IR$, j = l 2, ...,n

Meanwhile the membership *i* of the test sample is firstly unknown, we will define a matrix A for the entire training set as the concatenation of the n training samples of all k object classes:

$$A= [A_i,A_2, \ldots , A_k] = [v_{i,l}, v_{i,2},..., v_{k,nk}] \quad \text{(eq2)}$$

Formerly, the linear representation of y can be rewritten in terms of all training samples as

$$y = Ax_0 \in IR^m \quad \text{(eq3)}$$

here $x_0 = [0, --, 0, \alpha_{i,l,+} \alpha_{i,2+\ldots.+} \alpha_{i,ni}, 0,..., 0]^T \in IR^n$ is a coefficient vector whose entries are zero except those associated with the *i*th class.

In accordance with the entries of the vector x_0 coded the identity of the test sample y, it is tempting to try to obtain it by solving the linear system of equations y = A*x*. Notice, that using the whole training set to solve for *x* characterizes a most important departure from a sample or a class at a time methods such as NN and NS. Later we claim that one may achieve a more discriminative classifier from such a global representation. We have demonstrate its superiority over these local methods (NN or NS) both for categorizing objects represented in the training set and for discarding samples different class that do not arise from any of the classes existing in the training set. The benefits can come without an increase in the order of growth of the calculation: hence, the complexity remnants linear in the size of training set.

Clearly, if m > n, the system of equations y = A*x* is overdetermined, and the correct x_0 can usually be found as its unique solution. However, in robust face recognition, the system y = A*x* is usually underdetermined, and so, its solution is not unique. Conventionally, this problem can be resolved by selecting the minimumα ℓ^2 -norm solution:

$$(\ell^2): \quad \hat{x}_2 = \arg\min \|x\|_2 \text{ subject to } Ax = y. \quad \text{(eq4)}$$

Although this optimization problem can be simply resolved (via the pseudoinverse of A), the solution $\hat{x}_2$ is not specially informative for recognizing the test sample y. As presented in Example 1, $\hat{x}_2$ is in general dense, with large nonzero entries corresponding to training samples from numerous dissimilar classes. To solve this difficulty, we as an alternative exploit the following simple observation: A valid test sample y may be appropriately characterized using only the training samples from the same class. This representation is naturally sparse if the number of object classes k is reasonably enormous. For example, if k = 20, only 5 % of the entries of the desired x_0 should be nonzero. The sparser the recovered x_0, the easier will it be to precisely conclude the identity of the test sample y.

This will inspires us to look for the sparsest solution to y = Ax, solving the following optimization problem:

$(\ell 0): \quad \hat{x}_0 = \arg\min \|x\|_0 \text{ subject to } Ax = y, \quad (eq5)$

Here $\|.\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector. Actually, when the columns of A in general position, then at any time y = Ax for some x with less than m/2 nonzeros, x is the unique sparsest solution: $x_0 = x$. Though, the problem of finding the sparsest solution of an underdetermined system of linear equations is NP-hard and tough even to approximate: Usually, no known technique for finding the sparsest solution is significantly more effective than exhausting all subsets of the entries for x.

2.2 Sparse Solution via $\ell 1$-Minimization

Fig. 2. Geometry of sparse representation via $\ell 1$ -minimization. The ℓ^1-minimization determines which facet (of the lowest dimension) of the polytope A(Pα). The test sample vector y is represented as a linear combination of just the vertices of that facet, with coefficients x_0.That if the solution x_0 sought is sparse sufficient, the solution of the ℓ^0-minimization problem (eq5) is equal to the solution to the succeeding $\ell 1$ -minimization problem:

$(\ell 1): \quad \hat{x}_1 = \arg\min \|x\|_1 \quad \text{subject to } Ax = y. \quad (eq6)$

This difficulty can be resolved in polynomial time by standard linear programming methods. Even more effective procedures are available while the solution is known to be very sparse. For example, homotopy algorithms recover solutions with t nonzeros in O(t3 + n) time, linear in the size of the training set .

Geometric Interpretation

Geometric interpretation of minimizing the $\ell 1$ -norm correctly recovers sufficiently sparse solutions. Let Po denote the $\ell 1$ -ball (or crosspolytope) of radius α:

$Po=\{x : \|x\|1 < \alpha\} \quad C \quad IR^n. \quad (eq7)$

In the unit $\ell 1$ -ball P1 is mapped to the polytope P=A(P1) C IR^m, consisting of all y that satisfy y = Ax for some x whose $\ell 1$ -norm is < 1.

The geometric relationship between Pα and the polytope A(Pα) is invariant to scaling. That is, if we scale Pα, its image under multiplication by A is also scaled by the same amount. Geometrically, finding the minimum $\ell 1$ -norm solution x1 to (eq6) is equivalent to expanding the $\ell 1$ -ball P_α until the polytope A(Pα) first touches y. The value of o at which this occurs is precisely ||x||1.

At present, assume that $y = Ax_0$ for some sparse x_0. We want to know while solving (eq6) correctly recovers x0. This may easily resolved from the geometry Since x1 is found by expanding both Pα and A(Pα) until a point of A(Pα) touches y, the ℓ1 -minimizer x1 must generate a point Ax1 on the boundary of P.

Thus, x1 = x0 if and only if the point A(x0/||x0||1) lies on the boundary of the polytope P. For the example shown in Fig. 2, it is easy to see that the ℓ1 -minimization recovers all x0 with only one nonzero entry. This equivalence holds because all of the vertices of P1 map to points on the boundary of P.

In general, if A maps all t-dimensional facets of P1 to facets of P, the polytope P is referred to as (centrally) t-neighborly . From the above, we see that the ℓ1 -minimization (eq6) correctly recovers all x0 with < t + 1 nonzeros if and only if P is t-neighborly, in which case, it is Unfortunately, there is no known algorithm for efficiently verifying the neighborliness of a given polytope P. The best known algorithm is combinatorial, and therefore, only practical when the dimension m is moderate. When m is large, it is known that with overwhelming probability, the neighborliness of a randomly chosen polytope P is loosely bounded between

$$c \cdot m < t < [(m + 1)/3], \qquad \text{(eq8)}$$

for some small constant c > 0. Roughly speaking, if the number of nonzero entries of x0 is a small fraction of the dimension m, ℓ1 -minimization may recover x_o.

Dealing with Small Dense Noise

Before this, we have assumed that (eq3) holds exactly. Meanwhile real data are noisy, it would not be possible to express the test sample precisely as a sparse superposition of the training samples. The model (eq3) can be modified to explicitly account for small possibly dense noise by writing

$$y = Ax_o + z, \quad \text{(eq9)}$$

where $z \in IR^m$ is a noise term with bounded energy $||z||_2 < e$. The sparse solution x_o can still be approximately recovered by solving the following stable ℓ1 -minimization problem:

$$(\ell^1_s) : \hat{x}_1 = \arg\min ||x||_1 \text{ subject to } ||Ax — y||_2 < e. \quad \text{(eq10)}$$

This convex optimization problem can be efficiently solved via second-order cone programming. The solution of (ℓ1) is guaranteed to approximately recovery sparse solutions in ensembles of random matrices A : There are constants p and ζ such that with overwhelming probability, if $||xo||_o < pm$ and $||z||_2 < e$, then the computed x1 satisfies

$$||x1 — x0||_2 < \zeta\varepsilon \qquad \text{(eq11)}$$

Classification Based on Sparse Representation

Given a new test sample y from one of the classes in the training set, we first compute its sparse representation x1 via (eq6) or (eq10). Ideally, the nonzero entries in the estimate x1 will all be associated with the columns of A from a single object class i, and we can easily assign the test sample y to that class. However, noise and modeling error may lead to small nonzero entries associated with multiple object classes (see Fig. 3). Based on the global sparse representation, one can design many possible classifiers to resolve this. For instance, we can simply assign y to the object class with the single largest entry in x1. However, such heuristics do not harness the subspace

structure associated with images in face recognition. To better harness such linear structure, we instead classify y based on how well the coefficients associated with all training samples of each object reproduce y.

For each class i, let $\delta_i : IR^n \longrightarrow IR^n$ be the characteristic function that selects the coefficients associated with the lth class. For $x \in IR^n$, $\delta_i(x) \in IR^n$ is a new vector whose only nonzero entries are the entries in x that are associated with class i. Using only the coefficients associated with the ith class, one can approximate the given test sample y as $yl = A\,\delta_i(\hat{x}_1)$. We then classify y based on these approximations by assigning it to the object class that minimizes the residual between y and yl:

$$\min r_i(y) = \|y - A\,\delta_i\|(\hat{x}_1)\|_2. \qquad \text{(eq12)}$$

Algorithm 1 below summarizes the complete recognition procedure. Our implementation minimizes the ℓ1-norm via a primal-dual algorithm for linear programming based on and .

Algorithm 1. Sparse Representation-based Classification (SRC)
1: Input: a matrix of training samples
$A = [A_1, A_2, ..., A_k] \in IR^{mxn}$ for k classes, a test sample
$y \in IR^m$, (and an optional error tolerance $\varepsilon > 0$.)
2: Normalize the columns of A to have unit ℓ^2-norm.
3: Solve the ℓ^1-minimization problem:
$\hat{x}_1 = \arg\min_x \|x\|_1$ subject to $Ax = y$. (eq13)
(Or alternatively, solve $\hat{x}_1 = \arg\min_x \|x\|_1$ subject to $\|Ax - y\|_2 < \varepsilon$.)
4: Compute the residuals $r_i(y) = \|y - A\,\delta_i\|(\hat{x}_1)\|_2$ for i = 1,... ,k.
5: Output: identity (y) = $\arg\min_i r_i$ (y).
1 C1 -minimization versus ℓ2-minimization).

To illustrate how Algorithm 1 works, we randomly select half of the 2,414 images in the Extended Yale B database as the training set and the rest for testing. In this example, we subsample the images from the original 192 x 168 to size 12 x 10. The pixel values of the downsampled image are used as 120-D feature- s—stacked as columns of the matrix A in the algorithm. Hence, matrix A has size 120 x 1,207, and the system y = Ax is underdetermined. Fig. 3a illustrates the sparse coefficients recovered by Algorithm 1 for a test image from the first subject. The figure also shows the features and the original images that correspond to the two largest coefficients. The two largest coefficients are both associated with training samples from subject 1. Fig. 3b shows the residuals with respect to the 38 projected coefficients Si(X1), i = 1, 2,38. With 12 x 10 downsampled images as features, Algorithm 1 achieves an overall recognition rate of 92.1 percent across the Extended Yale B database. (See Section 4 for details and performance with other features such as Eigenfaces and Fisherfaces, as well as comparison with other methods.) Whereas the more conventional minimum ℓ2-norm solution to the underdetermined system y = Ax is typically quite dense, minimizing the ℓ1-norm favors sparse solutions and provably recovers the sparsest solution when this solution is sufficiently sparse. To illustrate this contrast, Fig. 4a shows the coefficients of the same test image given by the conventional ℓ2-minimization (eq4), and Fig. 4b shows the corresponding residuals with respect to the 38 subjects. The coefficients are much less sparse than those given by ℓ1 -minimization (in Fig. 3), and the dominant coefficients are not associated with subject 1. As a result, the smallest residual in Fig. 4 does not correspond to the correct subject (subject 1). 2.4Validation Based on Sparse Representation

Before classifying a given test sample, we must first decide if it is a valid sample from one of the classes in the data set. The ability to detect and then reject invalid test samples, or "outliers," is crucial for recognition systems to work in real- world situations. A face recognition system, for example, could be given a face image of a subject that is not in the database or an image that is not a face at all.
Systems based on conventional classifiers such as NN or NS, often use the residuals ri (y) for validation, in addition to identification. That is, the algorithm accepts or rejects a test sample based on how small the smallest residual is. However, each residual ri(y) is computed without any knowledge of images of other object classes in the training data set and only measures similarity between the test sample and each individual class.

In the sparse representation paradigm, the coefficients X1 are computed globally, in terms of images of all classes. In a sense, it can harness the joint distribution of all classes for validation. We contend that the coefficients X are better statistics for validation than the residuals. Let us first see this through an example.

concentration of sparse coefficients
We randomly select an irrelevant image from Google and downsample it to 12 x 10. We then compute the sparse representation of the image against the same Extended Yale B training data, as in Example 1. Fig. 5a plots the obtained coefficients, and Fig. 5b plots the corresponding residuals. Compared to the coefficients of a valid test image in Fig. 3, notice that the coefficients X here are not concentrated on any one subject and instead spread widely across the entire training set. Thus, the distribution of the estimated sparse coefficients X contains important information about the validity of the test image: A valid test image should have a sparse representation whose nonzero entries concentrate mostly on one subject, whereas an invalid image has sparse coefficients spread widely among multiple subjects.
For quantifying this observation, we define the following measure of how concentrated the coefficients are on a single class in the data set:
sparsity concentration index (SCI):
The SCI of a coefficient vector $x \in IR^n$ is defined as

$$SCI(\hat{x}) = (k \max_i \|x\|_1 / \|x\|_1)/(k-1) \in [0,1], \quad \text{(eq14)}$$

For a solution X found by Algorithm 1, if $SCI(\hat{x}) = 1$, the test image is represented using only images from a single object, and if $SCI(\hat{x}) = 0$, the sparse coefficients are spread evenly over all classes. We select a threshold $T \in (0,1)$ and accept a test image as valid if

$$SCI(\hat{x}) \geq T; \quad \text{(eq15)}$$

and otherwise reject as invalid. In step 5 of Algorithm 1, one may choose to output the identity of y only if it passes this criterion.

Unlike NN or NS, this new rule avoids the use of the residuals r (y) for validation. Notice, even for a non face image, with a large training set, the smallest residual of the invalid test image is not so large. Instead of relying on a single statistic for both validation and identification, our method separates the information requisite for these tasks: the residuals for identification and the sparse coefficients for validation. In this sense, the residual measures how good the representation approximates the test image; and the sparsity concentration index measures how good the representation itself is, in terms of localization.

One advantage to this method to validation is enhanced performance against generic objects those are similar to multiple object classes. For instance, in face recognition, a generic face could be somewhat similar to some of the subjects in the data set and may have small residuals regarding their training images. By means of residuals for validation may to be expected leads a false positive. Though, a generic face is not likely to pass the new validation rule as a good representation of it typically needs involvement from images of multiple subjects in the data set. Therefore, the new rule can better judge whether the test image is a generic face or the face of one specific subject in the data set. The new validation rule outperforms the NN and NS methods, with as much as 10-20 percent improvement in verification rate for a given false accept rate

Two Fundamental Issues in Face Recognition
we study the implications of general classification framework for two critical issues in face recognition: 1) the selection of feature transformation, and 2) robustness to corruption, occlusion, and disguise.

The Role of Feature Extraction
Numerous feature extraction schemes have been examined for finding projections that better distinct the classes in lower dimensional spaces, those are often referred to as feature spaces. A class of approaches extracts holistic face features such as Eigen-faces, Fisherfaces , and Laplacianfaces . Another class of approaches tries to extract meaningful partial facial features (e.g., patches around eyes or nose). Usually, when feature extraction is used in conjunction with simple classifiers such as NN and NS, the choice of feature transformation is considered critical to the success of the algorithm. This has led to the development of a wide variety of increasingly complex feature extraction methods, including nonlinear and kernel features , . In this section, we reexamine the role of feature extraction within the new sparse representation framework for face recognition.

One benefit of feature extraction, which carries over to the proposed sparse representation framework, is reduced data dimension and computational cost. For raw face images, the corresponding linear system y = Ax is very large. For instance, if the face images are given at the typical resolution, 640 x 480 pixels, the dimension m is in the order of 105. Although Algorithm 1 relies on scalable methods such as linear programming, directly applying it to such high-resolution images is still beyond the capability of regular computers.

Meanwhile utmost feature transformations involve only linear operations (or approximately so), the projection from the image space to the feature space may be represented as a matrix $R \in IR^{dxm}$ with d<<m sides of (eq3) yields

$$\tilde{y} = Ry = RAx_0 \in IR^d. \quad \text{(eq16)}$$

Actually, the dimension d of the feature space is normally selected to be much smaller than n. Here, the system of equations $\tilde{y} = RAx \in IR^d$ is underdetermined in the unknown $x \in IR^n$. Yet, as the desired solution x_o is sparse, we can hope to recover it by solving the following reduced $\ell 1$ -minimization problem:

$$(\ell^1_r): \hat{x}_1 = \arg\min_x ||x||_1 \text{ subject to } ||RAx - \tilde{y}||_2 \leq e, \quad \text{(eq17)}$$

for a given error tolerance e > 0. Thus, in Algorithm 1, the matrix A of training images is now replaced by the matrix

On behalf of existent face recognition approaches, experimental studies have shown that increasing the dimension d of the feature space usually improves the recognition rate, till the distribution of features RA_i does not turn into degenerate . Degeneracy is not an issue for $\ell 1$-minimization, because it merely requires that y be in or near the range of RA_i —it does not depend on the covariance $E = A^T{}_i R^T RA_i$ being nonsingular such as in classical discriminant analysis. The stable version of $\ell 1$ -minimization (eq10) or (eq17) is known in statistical literature as the Lasso. It efficiently regularizes highly underdetermined linear regression when the anticipated solution is sparse and has also been proven consistent in some noisy overdetermined settings.

For our sparse representation approach to recognition, we must have to understand how the selection of the feature extraction R affects the ability of the $\ell 1$ -minimization (eq17) to recover the correct sparse solution x_o. From the geometric explanation of $\ell 1$ -minimization, the answer to this be influenced by on whether the related new polytope P = RA(P;l) remains sufficiently neighborly. It is easy to show that the neighborliness of the polytope P = RA(P1) increases with d , . $\ell 1$ -minimization, in particular, the stable version (eq17), to recover sparse representations for face recognition using a variety of features. This proposes that most data- reliant features popular in face recognition (e.g., eigenfaces and Laplacian- faces) may indeed give highly neighborly polytopes P.

More analysis of high-dimensional polytope geometry may revealed a to a certain extent surprising phenomenon: if the solution x_o is sparse sufficient, then with overwhelming possibility, it may be appropriately recovered via $\ell 1$ -minimization from any sufficiently large number d of linear measurements $y = RAx_0$. More exactly, if x_0 has $t << n$ nonzeros, then with overwhelming probability

$$d \geq 2t \log(n/d) \quad \text{(eq18)}$$

random linear measurements are sufficient for $\ell 1$ -minimization (eq17) to recover the correct sparse solution x0 . This amazing occurrence has been dubbed the "blessing of dimensionality". Random features may be observed as a less-structured corresponding item to classical face features such as Eigenfaces or Fisherfaces. Consequently, we call the linear projection generated by a Gaussian random matrix Randomfaces.

The main benefit of Randomfaces is that they are extremely effective to generate, as the transformation R is autonomous of the training data set. This advantage would be significant for a face recognition system, where we cannot be able to attain a complete database of all subjects of concern to precompute data-dependent transformations for example Eigenfaces, or the subjects in the database may modified over time. In this cases, there is no necessity for re-computing the random transformation R.

Till the correct sparse solution x0 will be recovered, Algorithm 1 will every time give the same classification outcome, regardless of the feature actually used. Therefore, when the dimension of feature d go beyond the above bound (eq18), one would expect that the recognition performance of Algorithm 1 with different features rapidly converges, and the selection of an "optimal" feature transformation is no more critical: even if random projections or downsampled images would perform well like other wisely engineered features.

Robustness to Occlusion or Corruption

In numerous real-world face recognition circumstances, the test image y might be partially corrupted or occluded. In this situation, the overhead linear model (eq3) must be adapted as

$$y = y_0 + e_0 = A\, x_0 + e_0, \qquad \text{(eq19)}$$

Where $e_0 \in IR^m$ is a vector of errors—a portion, p, of its entries are nonzero. The nonzero entries of e_0 model which pixels in y are degraded or occluded. The positions of corruption may be different for dissimilar test images and are not known to the computer. The errors can poses random magnitude and thus cannot be ignored or treated with procedures designed for small noise.

Basic principle of coding theory is redundancy in the measurement is essential to sensing and fixing gross errors. Redundancy arises in object recognition for the reason that the number of image pixels is usually far greater than the number of subjects that have created the images. In that situation, even if a fraction of the pixels are entirely corrupted by occlusion, recognition would yet be possible using the remaining pixels. Alternatively, feature extraction schemes deliberated in the previous section may reject valuable information that would support pay off for the occlusion. In this way, no representation is more redundant, robust, or informative than the original images. Thus, when dealing with occlusion and corruption, we must always work with the maximum possible resolution, carrying out downsampling or feature extraction so long as the resolution of the original images is too high to process.

Obviously, redundancy should be of no use short of efficient computational tools for take advantage of the info coded in the redundant data. The trouble in directly harnessing the redundancy in degraded raw images has led researchers to as an alternative focus on spatial locality as a guiding principle for strong recognition. Local features work out from simply a small fraction of the image pixels are clearly less likely to be corrupted by occlusion than holistic features. Face recognition, approaches such as ICA and LNMF exploit this observation by adaptively selecting filter bases that are locally focused. Local Binary Patterns and Gabor wavelets have analogous properties, meanwhile they are also computed from local image regions. An interrelated method partitions the image into fixed regions and computes features for each region. Notice, and yet, that projecting onto locally focused bases transforms the domain of the occlusion problem, rather than excluding the occlusion. Mistakes on the original pixels become errors in the transformed domain and can even become less local. The role of feature extraction in accomplishing spatial locality is therefore disputed, since no bases or features can more spatially localized than the original image pixels themselves. Actually, the most widespread approach to robustifying feature- based methods is based on arbitrarily sampling individual pixels, sometimes in conjunction with statistical methods such as multivariate trimming.

Now, let us express how the proposed sparse representation classification framework may be extended to treat with occlusion. Let us assume that the degraded pixels are a comparatively lesser portion of the image. The error vector e_0, like the vector x_0, then has sparse nonzero entries. Since $y_0 = Ax_0$, we can rewrite (eq19) as

$$y = [A, I]\begin{bmatrix} x0 \\ e0 \end{bmatrix} = Bw_0$$

Here, $B = [A, I] \in IR^{mx(n+m)}$, so the method y = Bw is always underdetermined and does not have a unique solution for w. On the other hand, from the above discussion

about the sparsity of x_0 and e_0, the exact producing $w_0 = [x_0, e_0]$ has at most $n_i + \rho m$ nonzeros. We might then expect to recover w_0 as the sparsest solution to the system $y = Bw$. In fact, if the matrix B is in general position, then as long as $y = Bw$ for some w with less than m/2 nonzeros, $\tilde{w}$ is the unique sparsest solution. Thus, if the occlusion e covers less than $(m-n_i)/2$ pixels, approximately 50 % of the image, the sparsest solution w to $y = Bw$ is the true generator, $wo = [x_0, e_0]$.

More commonly, we assume that the corrupting error e_0 has a sparse representation wrt some basis $A_e \in IR^{mxne}$. That is, $e_0 = A_e u_o$ for some sparse vector $u_0 \in IR^m$. Now, we have chosen the special case $A_e = I \in IR^{mxm}$ as e_0 is supposed to be sparse with respect to the natural pixel coordinates. If the error e_0 is as an alternative more sparse with respect to another basis, e.g., Fourier or Haar, we can easily re-define the matrix B by appending A_e (instead of the identity I) to A and instead seek the sparsest solution w_0 to the equation:

$$y = Bw \text{ with } B = [A, A_e] \quad \in IR^{mx(n+ne)}. \quad \text{(eq21)}$$

By this approach, the similar formulation can handle more general classes of (sparse) corruption.

As already, we try to recover the sparsest explanation w_0 from solving the following extended ℓ^1-minimization difficulty:

$$(\ell^1_e): \quad \hat{w}_1 = \arg\min \|w\|_1 \quad \text{dependent on } Bw=y. \quad \text{(eq22)}$$

That is, in Algorithm 1, we now replace the image matrix A with the extended matrix B = [A, I] and x with w = [x, e].

Obviously, whether the sparse solution w_0 may be recovered from the above ℓ^1-minimization be influenced by the neighborliness of the new polytope $P = B(P_l) = [A, I](P_l)$. This polytope comprises vertices from both the training images A and the identity matrix I. The bounds specified in (eq8) indicate that if y is an image of subject i, the ℓ^1-minimization (eq22) cannot promise to correctly recover $w_0 = [x_0, e_0]$ if

$$n_i + |\text{support}(e_0)| > d/3.$$

Usually, $d >> n_i$, subsequently, (eq8) suggests that the highest fraction of occlusion in which we may expect to still achieve exact reconstruction is 33%. The bound is supported by our experimental results.

For knowing exactly to what extent occlusion can be endured, we need more precise information about the neighborliness of the polytope P than a loose upper bound given by equation (eq8). To instance, we be interested to know for a given set of training images, what is the biggest amount of (worst possible) occlusion it can tolerate. Although the best known algorithms for precisely figuring the neighborliness of a polytope are combinatorial in nature, close-fitting upper bounds can be acquired by constraining the search for intersections between the null space of B and the ℓ^1-ball to a random subset of the t-faces of the ℓ^1-ball. We use this method to estimate the neighborliness of all the training data sets considered in our experiments.
Empirically, we got that the stable form (eq10) is only essential if we do not consider occlusion or corruption e_0 in the model. After we explicitly account for gross errors by using B = [A, I] the extended ℓ^1 -minimization (eq22) with the precise limit Bw = y is already stable within moderate noise.

As soon as the sparse solution $\hat{w}_i = [\hat{x}_1, \hat{e}_1]$ is figured out, setting $y_r = y - \hat{e}1$ recovers a clear image of the object with occlusion or corruption compensated for. To recognize the subject, we slightly change the residual $r_i(y)$ in Algorithm 1, calculating it against the recovered image y_r:

$$r_i(y) = \|y_r - A\delta_i(\hat{x}_1)\|_2 = \|y - \hat{e}_1 - A(\hat{x}_1)\|_2 \qquad \text{[eq23]}$$

CHAPTER-6

EXPERIMENTS AND RESULT

6.1. EXPERIMENT WITH YALE B DATABASE

In this experiment I have used the Yale B database. Yale database contains 38 different person each person have 64 images. In this experiment we use 3 training images and all images as the testing image. For further increase the accuracy we may increase the training image but with increase in training image recognition time may also be increased so I have just set the no of training image to 3

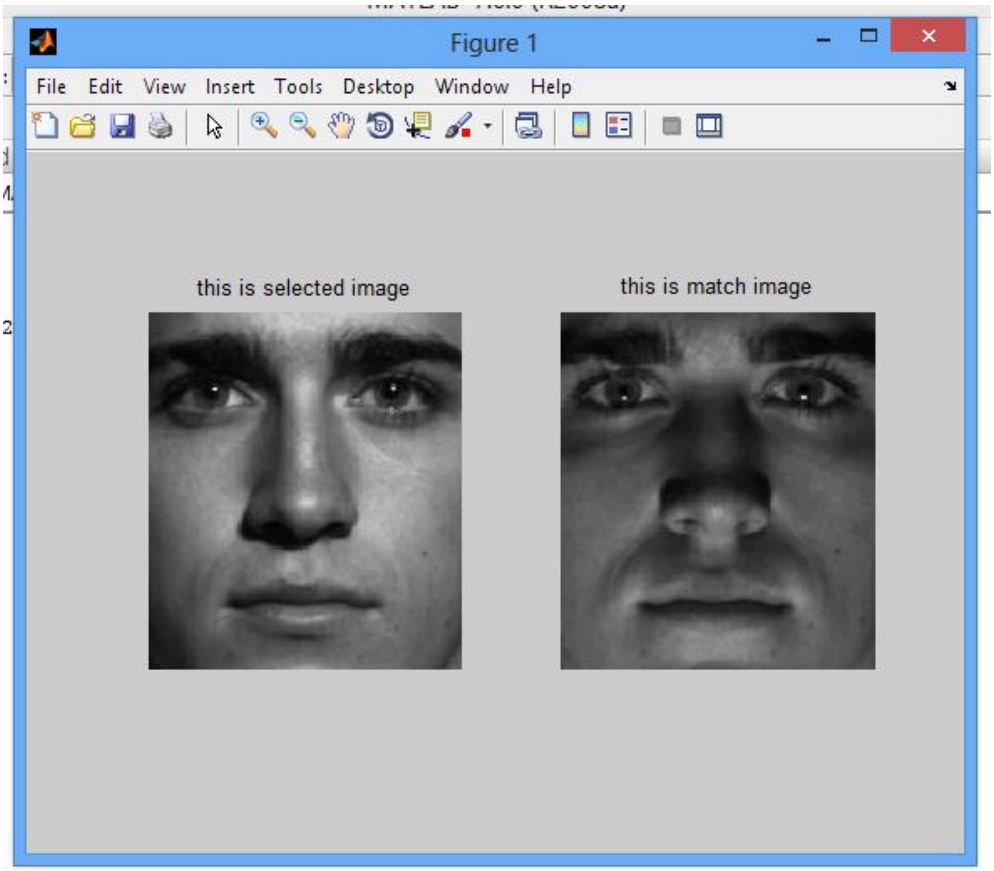


Figure 1: it shows the recognition in poses changing condition

The above face recognition is in the condition of the varying pose condition first image is the person to be recognized second image is the recognized image. The above result of the experiment shows that the algorithm works in pose changing conditions.

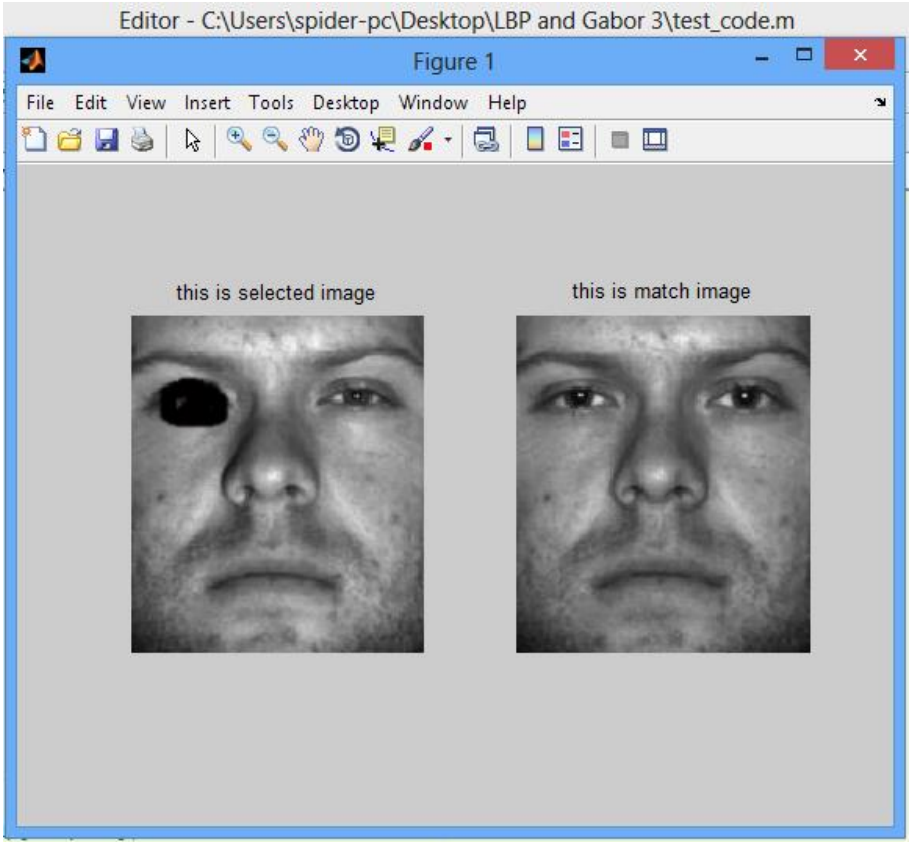


Figure 2: the above figure shows the recognition in condition of occlusion

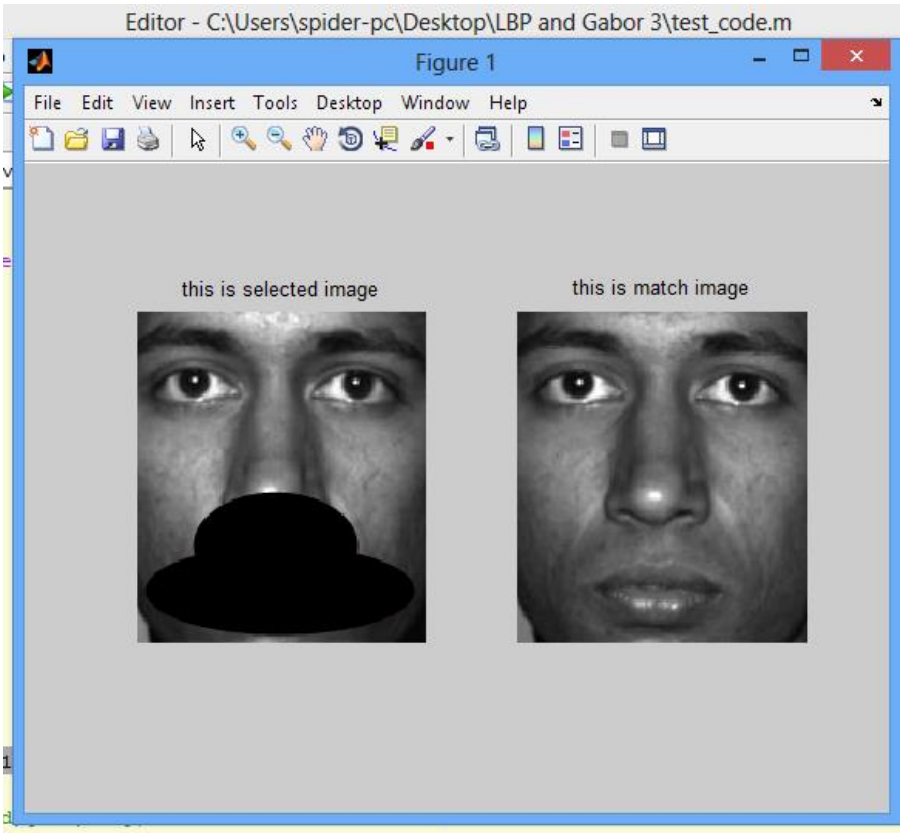


Figure 3: case of disguise to be recognized by proposed model

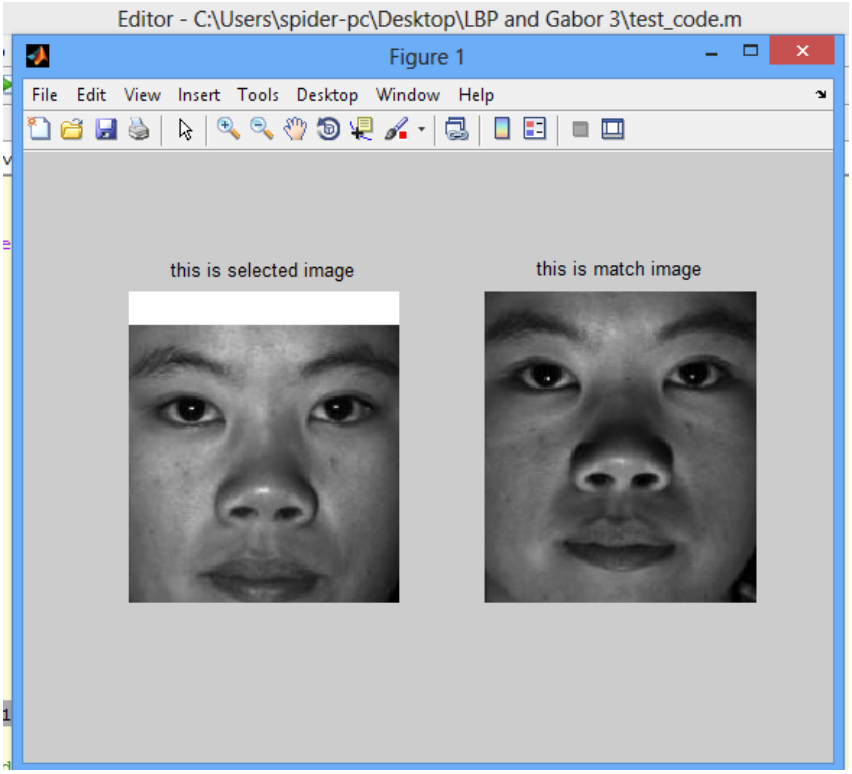


Figure 4: condition of misaligned Face recognition

6.2. EXPERIMENT WITH ORL DATABASE

Now we are checking the efficiency on the ORL database it have 40 person every person have 10 images we use limited database 20 person. We have used 1-6 as training image and 7-10 as testing image.

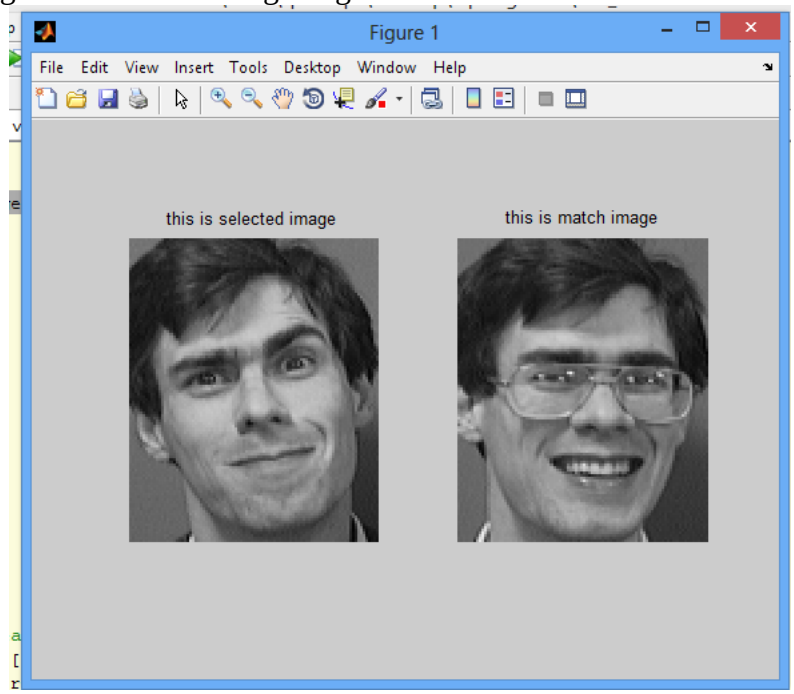


Figure shows person recognized with different pose

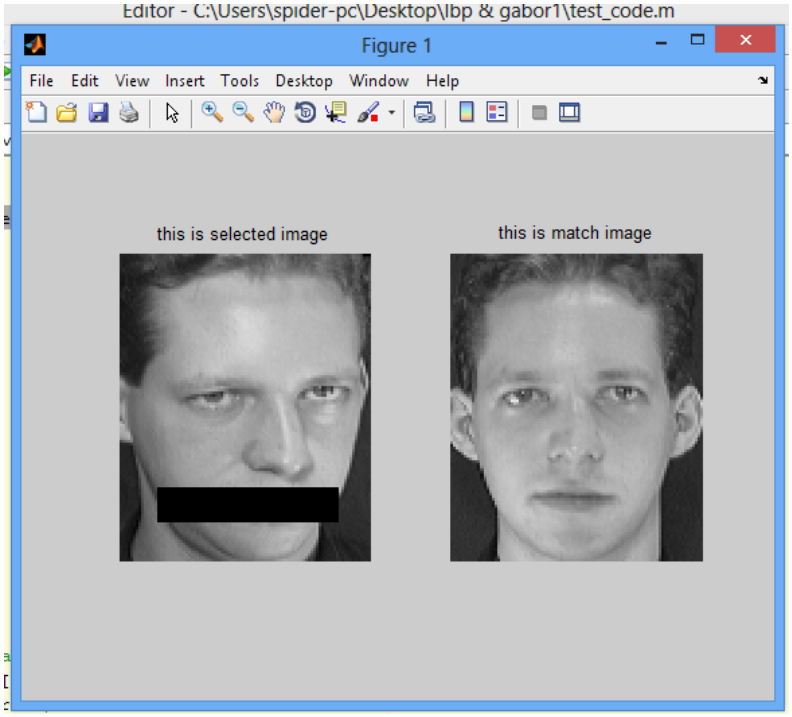


Figure shows occluded face recognized by the system

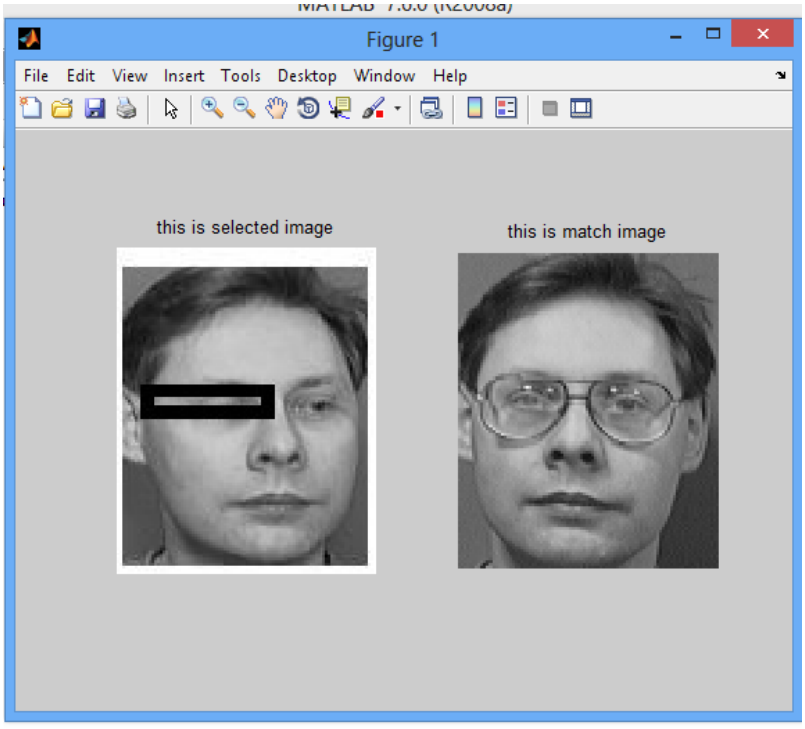
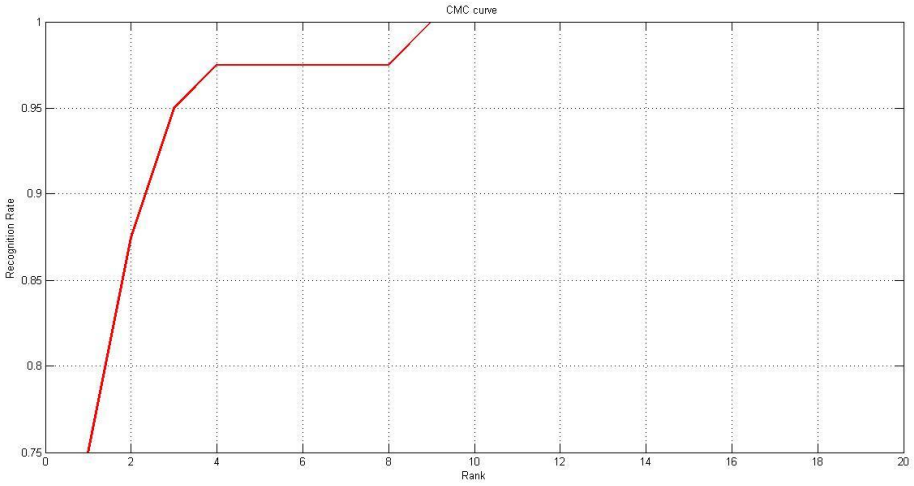
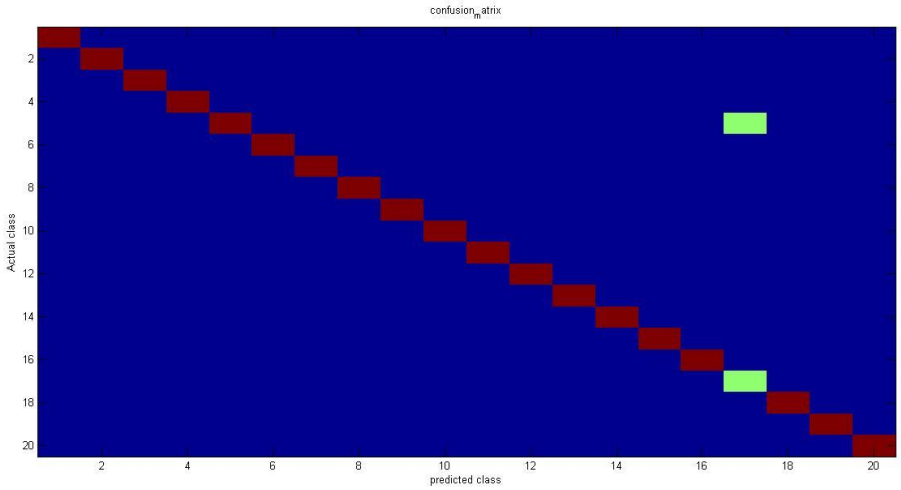


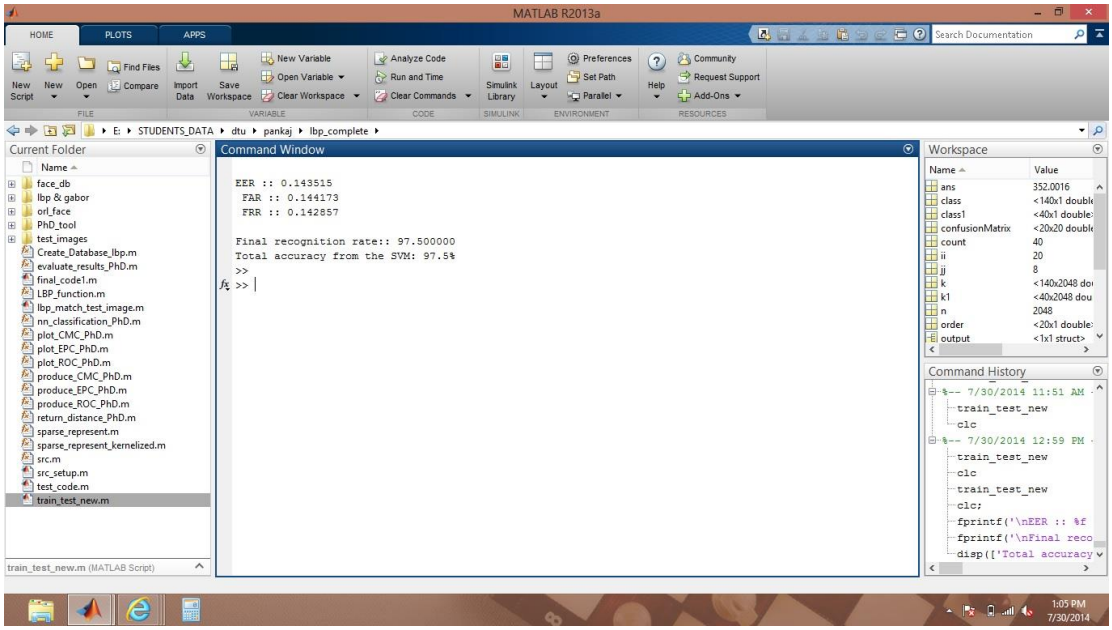
Figure shows the recognization under unaligned ocluded and disguied image recognition simultanuously



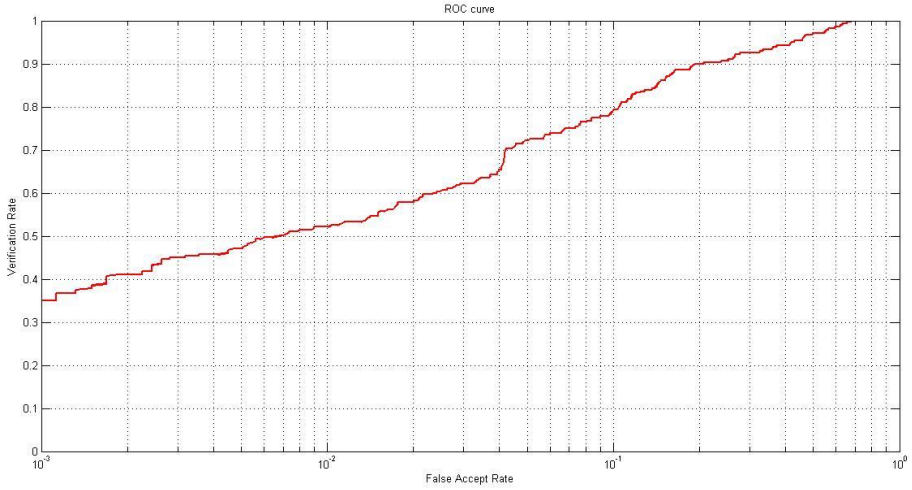
CMC Curve



Confusion Matrix



Result



Roc Curve

CHAPTER-7

CONCLUSION AND FUTURE WORK

In this report, I tried to check experimentally that exploiting sparsity is critical for the high-performance classification of high-dimensional data such as face images. With sparsity properly harnessed, the choice of features becomes less important than the number note of features used (in our face recognition example, approximately 100 are sufficient to make the difference negligible). However the choice of feature may improve the accuracy and efficiency of the system.

Besides, occlusion and corruption may be handled uniformly and robustly within the same classification framework. One can achieve a striking recognition performance for severely occluded or corrupted images by a simple algorithm with no special engineering.

Due to the vast verities of problem can be handled by using the sparse representation approach in this technique for example Illumination variation, pose change and expression variation, Disguise condition We can develop systems for different purposes related to the real time such as for video surveillance, image searching for image on internet, robotics, industries by using sparse representation classifier.

Video Surveillance

Video surveillance can be done to capture a terrorist at crowdie places i.e. airports, train stations, bus stands, boarder of the country. It is not possible for human to remember to number of faces by only seeing picture for criminal more over it is also not possible to identify a face in large crowd And also it is not possible for government to hire too many employee to watch each and every corner of the place at same time due to the limited perception area and identifying speed because of the above mentioned problems the efficiency of the video surveillance can be achieved only by using the automatic surveillance systems.

Image searching on internet

This may be special application in which the person image can be searched on internet by image available instead of name. this may help to establish the system by which required person can be find out on the internet whether he change his name or not it will greatly help in finding out the persons those are doing fraud using the internet. Using many identities such as recently find out the many voters are find out using more than one identity and voting more than one time. Criminals may be find out using bank account with different names managed by the same person for hiding the black money collected by him by doing unfair works. To deceiving the police, income tax department and other government related organization.

Robotics

From many decades human is beyond developing robots and robotic machines. The invention of these machines is incomplete without the invention of the proper

identification system because the robots are those machines which can do work in replacement to the human this can be achieved if and only robots can identify the human perfectly as humans do in different conditions and uncontrolled scenarios due to robots are developed for working the real word to replace the human and in the real world conditions are always different for the different times and different places. For example day light conditions different from the night conditions, room conditions different from outside conditions, rainy conditions are different from the clear weather conditions.

Industries and Companies

In industries and companies some special secretes and area are available which must only be accessible only by the top level person or entry of the other persons may be permitted only with the help of the top level manage for example in bank cash is retained at the area where entry of other staff member is permitted only with the help of the manager this is a problem in the area called bio-metric.