FACE RECOGNITION USING DIFFERENT STATISTICAL FEATURES

A thesis submitted toward partial fulfillment
of the requirements for the degree of
Master of Technology in
IT(Courseware Engineering)
Jadavpur University

Submitted by
TIRTHARAJ SAMAJPATI
ROLL NO : M48CWE14-06

Under the guidance of
Dr. Ranjan Parekh
Jadavpur University

School Of Education Technology
M.TECH in IT(CWE) course affiliated
to Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032
India

2014

M.TECH. in IT(Courseware Engg) course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled "**FACE RECOGNITION USING DIFFERENT STATISTICAL FEATURES** " is a bonafide work carried out by TIRTHARAJ SAMAJPATI under our supervision and guidance for partial fulfilment of the requirement for Post Graduate Degree of Master in Technology in IT(Courseware Engineering) during the academic session 2013-2014.

THESIS ADVISOR
Dr. Ranjan Parekh ,Professor
Jadavpur university,
Kolkata-700 032

DIRECTOR
School of Education Technology,
Jadavpur University,
Kolkata-700 032

DEAN
Faculty Council of Interdisciplinary Studies, Law and Management
Jadavpur University,
Kolkata-700 032

CERTIFICATE OF APPROVAL **

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactorily to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

Committee of final examination --

for evaluation of Thesis

** Only in case the thesis is approved.

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of his M.TECH IT(COURSEWARE ENGINEERING) studies.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name : TIRTHARAJ SAMAJPATI

Exam Roll Number : M48CWE14-06

Thesis Title : FACE RECOGNITION USING DIFFERENT STATISTICAL FEATURES

Signature with Date:

ACKNOWLEDGEMENT

It is my greater pleasure to be able to thank those people without whose assistance my dissertation would not have been possible. Without the guidance and immeasurable assistance of these people I wouldn't have reached this phase. Firstly I would like to express my sincere gratitude to my guide, **Dr. Ranjan Parekh** for his assistance and guidance without which this work wouldn't have been completed. His timely advices are the key to my success. I am also grateful to **Prof. Samar Bhattacharya**, Director of School of Education technology for providing me his valuable opinions, suggestions and support. I am really indebted to **Dr. Matangini Chattopadhyay**, **Mr. Arunasish Acharya, Mr. Joydeep Mukherjee** and **Mrs. Saswati Mukherjee** for their continuous support and valuable advice during the entire course of the project.

I am also thankful to my classmates of Master in Multimedia Development and M.Tech (IT Courseware Engg.) for supporting me from time to time and providing me with assistance whenever I needed them. I am also thankful to our departmental staffs and all those who were associated with the project directly or indirectly. Finally, I would like to thank to my family members for their understanding, support, inspiration, encouragement and endless love through the duration of my project.

Regards,

TIRTHARAJ SAMAJPATI
Class Roll No: 001230402011
Exam Roll No: M48CWE14-06
Date: Registration Number: 121350 of 2012-13
M.TECH IT(COURSEWARE ENGG)
School of Education Technology,
Jadavpur University, Kolkata- 700032

CONTENTS

LIST OF TABLES

Table No. and Name **Page No.**

LIST OF IMAGES

Image No. and Name **Page No.**

Executive summary

This paper describes the design, implementation, and evaluation of a research work for developing a face recognition system using different statistical features. The developed face recognition system mainly uses RGB image analysis provided by MATLAB environment. In order to train and test the developed face recognition system, a dataset is used, which contains face images of 50 persons each of which is repeated 7 times. Therefore, a total of 350 face images are collected. The collected face images have passed through pre-processing steps for training and testing the face recognition system. Features such as wavelet decomposition, correlation coefficient, entropy and mean from histogram are selected to be used in the system, which reflects information about the face images. Manhattan distance classifier is used to classify the test images. Overall, the face recognition system has obtained an average recognition rate of 93.33% for 50 persons.

CHAPTER 1

INTRODUCTION

One of the most common biometrics recognition is face recognition. The face recognition has attracted many researchers' attention in the recent years, since the human face offers a relatively easy and reliable way to identify individuals, compared to other identification means. Some existing applications that employ face recognition include mug shot matching, credit card verification, ATM access, home PC access, video surveillance, etc .The first step involved in an automated face recognition is the face detection by which the location and size of each face is determined. The reliability of the face detection process has a major influence on the performance and usability of the whole face recognition application. Face Recognition is a form of biometric identification .A biometrics is,“ Automated methods of recognizing an individual based on their unique physical or behavioral characteristics“. The process of Facial recognition involves automated methods to determine identity, using facial features as essential elements of distinction. Three types of feature extraction methods can be distinguished:

- Generic methods based on edges, lines, and curves;
- Feature-template-based methods that are used to detect facial features such as eyes;
- Structural matching methods that take into consideration geometrical constraints on the features.

These methods have difficulty when the appearances of the features change significantly, for example, closed eyes, eyes with glasses, open mouth.

These methods are classified into three main categories:

1. Holistic matching methods,
2. Feature-based matching methods and
3. Hybrid methods.

- **Holistic matching** methods use the whole face region as the raw input to a recognition system. Using PCA, many face recognition techniques have been developed: Eigen faces, feature-line based methods, Fisher faces, Bayesian methods and SVM methods.

- In **Feature-based matching** methods, local features such as the eyes, nose, and mouth are first extracted and their locations and local statistics (geometric and/or appearance) are fed into a structural classifier.
- **Hybrid methods** are based on using both local features and the whole face region to recognize a face, as the human perception system uses.

1.1CHALLENGES

Face recognition is a very interesting quandary. Ideally a face detection system should be able to take a new face and return a name identifying that person. Mathematically, what possible approach would be robust and fairly computationally economical? If we have a database of people, every face has special features that define that person. Greg may have a wider forehead, while Jeff has a scar on his right eyebrow from a rugby match as a young tuck. One technique may be to go through every person in the database and characterize it by these small features. Another possible approach would be to take the face image as a whole identity.

Statistically, faces can also be very similar. Walking through a crowd without glasses, blurry vision can often result in misidentifying someone, thus yielding an awkward encounter. The statistical similarities between faces give way to an identification approach that uses the full face. Using standard image sizes and the same initial conditions, a system can be built that looks at the statistical relationship of individual pixels. One person may have a greater distance between his or her eyes then another, so two regions of pixels will be correlated to one another differently for image sets of these two people.

From a signal processing perspective the face recognition problem essentially boils down to the identification of an individual based on an array of pixel intensities. Using only these input values and whatever information can be gleaned from other images of known individuals the face recognition problem seeks to assign a name to an unknown set of pixel intensities.

Characterizing the dependencies between pixel values becomes a statistical signal processing problem. The Eigen face technique finds a way to create ghost-like faces that represent the majority of variance in an image database. Our system takes advantage of these similarities between faces to create a fairly accurate and computationally "cheap" face recognition system.

1.2. PROBLEM STATEMENT

Face Recognition using different statistical features

This thesis deals with face classification for different classes based on the features directly extracted from the faces by using image processing technique.

1.3. OBJECTIVES

- To study the existing methods for recognize the faces of different persons.
- To investigate how to improve the existing methods of face recognition.
- To develop an implemented recognition technique from the existing techniques studied.

CHAPTER 2

LITERATURE SURVEY

Various techniques for face recognition have been used in different context. One of the earlier works in [1] the author proposes DBC based Face Recognition using DWT (DBC-FR) model. The images in the database and test image are processed before extracting the features. Preprocessing involves, (i) Color to gray scale image conversion; the gray scale image with intensity values between 0 and 255 is obtained from color image to reduce processing time (ii) Image cropping; face image contains background and other occlusions that may not be required to identify a person correctly. Hence, only the face portion of the image is cropped. The image is converted to binary prior to cropping using the threshold value and pixel value. For the first portion, image is scanned from left to right until a binary 1 is encountered. When binary 1 is obtained, scanning of that row is stopped by storing the pixel's position and it is repeated for all the rows. This procedure is repeated for the second half portion scanning the image from right to left, then image is cropped based on stored pixel's positions. (iii) Image resizing; the cropped database and test image may have different dimensions, hence the cropped image is resized to a uniform 100*100 dimension. Wavelet transform is a powerful mathematical tool used to extract localized time-frequency (spatial-frequency) information of an image. The decomposition of the data into different frequency ranges is made using mother (prototype) wavelet and scaling functions or father wavelets and is reversible in its operation. CWT uses all possible scales and translations, where as DWT use only selected. By applying DWT, the image is decomposed into approximation and detail components namely Low-Low (LL), High-Low (HL), Low-High (LH), High-High (HH) sub bands, corresponding to approximate, horizontal, vertical, and diagonal features respectively. The dimension of each sub band is half the size of original image. The sub bands HL and LH contain the changes of images or edges along horizontal and vertical directions respectively. The HH sub band contains high frequency information of the image. LL band is considered as the most significant information of the face image. The process of face recognition involves identifying the person based on facial features. Hence it is necessary to extract the features from a given image. DBC is applied on LL sub band to encode the directional edge information. It captures the spatial relationship between any pair of neighborhoods pixels in a local region along a given direction. It reflects the image local feature. Euclidean Distance (ED) is used to verify whether the person is in database or not. If the ED value is less than the threshold, we have to check whether the person from the database and test image of a person is same. If it is same, then the match count is incremented, else the mismatch count is incremented. If the ED value is greater than threshold then the false rejection rate count is incremented indicating the image in database is falsely rejected.

In [2] the authors proposed an integrated image fusion and match score fusion of multispectral face images. The fusion of visible and long wave infrared face images is performed using 2_Granular SVM which uses multiple SVMs to learn both the local and global properties of the multispectral face images at different granularity levels and resolution. The 2_GSVM performs accurate classification which is subsequently used to dynamically compute the weights of visible and infrared images for generating a fused face image. 2D log polar Gabor transform and local binary pattern feature extraction algorithms are applied to the fused face image to extract global and local facial features respectively. The corresponding match scores are fused using Dezert Smarandache theory of fusion which is based on plausible and paradoxical reasoning. The efficacy of the proposed algorithm is validated using the Notre Dame and Equinox databases and is compared with existing statistical, learning, and evidence theory based fusion algorithms.

In [3] the authors proposed a novel approach for matching low resolution probe images with higher resolution gallery images, which are often available during enrollment, using Multidimensional Scaling (MDS). The ideal scenario is when both the probe and gallery images are of high enough resolution to discriminate across different subjects. The proposed method simultaneously embeds the low-resolution probe images and the high-resolution gallery images in a common space such that the distance between them in the transformed space approximates the distance had both the images been of high resolution. The two mappings are learned simultaneously from high-resolution training images using an iterative majorization algorithm. Extensive evaluation of the proposed approach on the Multi-PIE data set with probe image resolution as low as 8 _ 6 pixels illustrates the usefulness of the method. We show that the proposed approach improves the matching performance significantly as compared to performing matching in the low-resolution domain or using super-resolution techniques to obtain a higher resolution test image prior to recognition. Experiments on low-resolution surveillance images from the Surveillance Cameras Face Database further highlight the effectiveness of the approach.

In [4] the authors proposed Principle Component Analysis PCA is a classical feature extraction and data representation technique widely used in pattern recognition. It is one of the most successful techniques in face recognition. But it has drawback of high computational especially for big size database. This paper conducts a study to optimize the time complexity of PCA (eigenfaces) that does not affects the recognition performance. The authors minimize the participated eigenvectors which consequently decreases the computational time. A comparison is done to compare the differences between the recognition time in the original algorithm and in the enhanced algorithm. The performance of the original and the enhanced proposed algorithm is tested on face94 face database. Experimental results show that the recognition time is reduced by 35% by applying our proposed enhanced algorithm. DET Curves are used to illustrate the experimental results.

In [5] the authors proposed a multi-resolution feature extraction algorithm for face recognition is proposed based on two-dimensional discrete wavelet transform (2D-DWT), which efficiently exploits the local spatial variations in a face image. For the purpose of feature extraction, instead of considering the entire face image, an entropy-based local band selection criterion is developed, which selects high-informative horizontal segments from the face image. In order to capture the local spatial variations within these high informative horizontal bands precisely, the horizontal band is segmented into several small spatial modules. Dominant wavelet coefficients corresponding to each local region residing inside those horizontal bands are selected as features. In the selection of the dominant coefficients, a threshold criterion is proposed, which not only drastically reduces the feature dimension but also provides high within-class compactness and high between-class separability. A principal component analysis is performed to further reduce the dimensionality of the feature space. Extensive experimentation is carried out upon standard face databases and a very high degree of recognition accuracy is achieved by the proposed method in comparison to those obtained by some of the existing methods.

In [6] the authors proposed (i) Skin Region Detection, (ii) Face Candidate Localization. The algorithm initially estimates and projects the lighting compensated red component (LCRC) content of the image. The LCRC component is evolved from the input color image by transforming the RGB Color Space component of the image onto YCbCr Color Space component and removing the intensity and the blue component from the image. The resulting LCRC image in the YCbCr color space is remodelled back to the RGB color space. Only the Red component is retained in the resulting image with respect to the fact that the human skin is rich in red component because of the presence of the blood irrespective of the color of the skin. The modified image is subjected to skin detection algorithm which detects only the true skin regions in the image. It is evident from the experimental results that fallacious skin region detection is trounced proficiently. The proposed face detection algorithm performs efficiently. The algorithm detects the valid skin regions and projects the face region out of the detected skin regions. The complexity involved in computation is relatively more proficient when compared to that of the prior developed methodologies because of the fact that the luminance information is excluded from the computation. The pixel and the region based valid skin color extraction technique are applied only on the bespoken chrominance information.

In [7] the authors proposed a meaningful representation and an effective recognition method of color images in a unified framework. The author integrates color image representation and recognition into one discriminant analysis model: color image discriminant (CID) model. In contrast to the classical FLD method, which involves only one set of variables (one or multiple discriminant projection basis vectors), the proposed CID models involve two sets of variables: a set of color component combination coefficients for color image representation and one or multiple discriminant projection basis vectors for image discrimination.

The two sets of variables can be determined optimally and simultaneously by the proposed CID algorithms. For easy understanding and derivation, the author first presents a basic CID (BCID) model for two-class recognition problems. The BCID model contains one color component combination coefficient vector and one discriminant projection basis vector. The author uses the Lagrange multiplier method to solve the optimization problem that the BCID model involves and design a BCID algorithm to seek the optimal solution by solving two associated, generalized Eigen equations iteratively. The author then develops a general CID (GCID) model and an algorithm for multiclass recognition problems. The GCID algorithm can determine an optimal discriminating color component and a set of optimal discriminant projection basis vectors. Because one discriminating color component is generally not enough for the discrimination of color images, the GCID algorithm is further extended to generate multiple discriminating color components (such as the three color components of the RGB color images) for further improving the color image recognition performance. The author uses the face recognition grand challenge (FRGC) database and the biometric experimentation environment (BEE) system to assess the proposed CID models and algorithms. FRGC is the most comprehensive face recognition efforts organized so far by the U.S. Government, and it consists of a large amount of face data and a standard evaluation system, known as the BEE system. The BEE baseline algorithm reveals that the FRGC version 2 Experiment 4 is the most challenging experiment, because it assesses face verification performance of controlled face images versus uncontrolled face images. The author, therefore, chooses FRGC version 2 Experiment 4 to evaluate his algorithms. The author implements three experiments: one for gender recognition and two for face verification, and the experimental results demonstrate the effectiveness of the proposed models and algorithms.

In [8] the authors proposed face recognition framework based on person-specific SIFT features in three parts: Firstly, each input face image is normalized and extracted with SIFT features; Secondly, a k-means clustering on the locations of features is computed to construct sub-regions in face images; Thirdly, a matching computation is processed between a testing image and all registered images for recognizing face. Scale Invariant

Feature Transform has been proposed for extracting distinctive invariant features from images to perform matching of different views of an object or scene. It consists of two main parts: interest point detector and feature descriptor. The author proposes to ensemble a K-means clustering scheme to construct the sub-regions automatically based on the locations of features in training samples. The clustering scheme is as follows:(1) For input registered images, initialize k sub-region cluster centers with random values.(2) Decide the nearest sub-region for each feature point in each image using the Euclidean distance and update the values of each center to reconstruct the sub-regions.(3) If the new centers remain the same as before recomputed, stop clustering and the remaining k sub regions are the resulting areas for matching.(4) After constructing the sub-regions on face image, when testing a new image, all the SIFT features extracted from the image are assigned into corresponding sub regions based on the locations. SIFT features perform quite well and robust even under a single training image. However, the method fails to work under lighting and age variations because of the person-specific features may be more sensitive to these variations.

In [9] the authors proposed a novel approach for face recognition using moments. Four feature extraction methods have been used: Hu moments, Zernike moments, Legendre moments and Cumulants. Hu moments include a set of seven moments which are derived from the conventional geometric moments. These moments are invariant against rotation, scaling and translation. Legendre and Zernike moments have an orthogonal basis set and can be used to represent an image with minimum amount of information redundancy. These are based on the theory of orthogonal polynomials and can be used to recover an image from moment invariants. Cumulants are sensitive to image details and therefore are suitable for representing the image features. For feature extraction, moments of different orders are calculated which form the feature vectors. The feature vectors obtained are stored in the database and are compared using three different classifiers. In case of cumulants, we have calculated the bispectrum of images and compressed it using wavelets.

In [10] the authors proposed a face-mosaicing scheme that generates a composite face image during enrollment based on the evidence provided by frontal and semi profile face images of an individual. Face mosaicing obviates the need to store multiple face templates representing multiple poses of a user's face image. In the proposed scheme, the side profile images are aligned with the frontal image using a hierarchical registration algorithm that exploits neighborhood properties to determine the transformation relating the two images.

Multiresolution splining is then used to blend the side profiles with the frontal image, thereby generating a composite face image of the user. A texture-based face recognition technique that is a slightly modified version of the C2 algorithm proposed by Serre et al. is used to compare a probe face image with the gallery face mosaic. Experiments conducted on three different databases indicate that face mosaicing, as described in this paper, offers significant benefits by accounting for the pose variations that are commonly observed in face images.

CHAPTER 3

PROPOSED APPROACH

3.1. *BLOCK DIAGRAM*

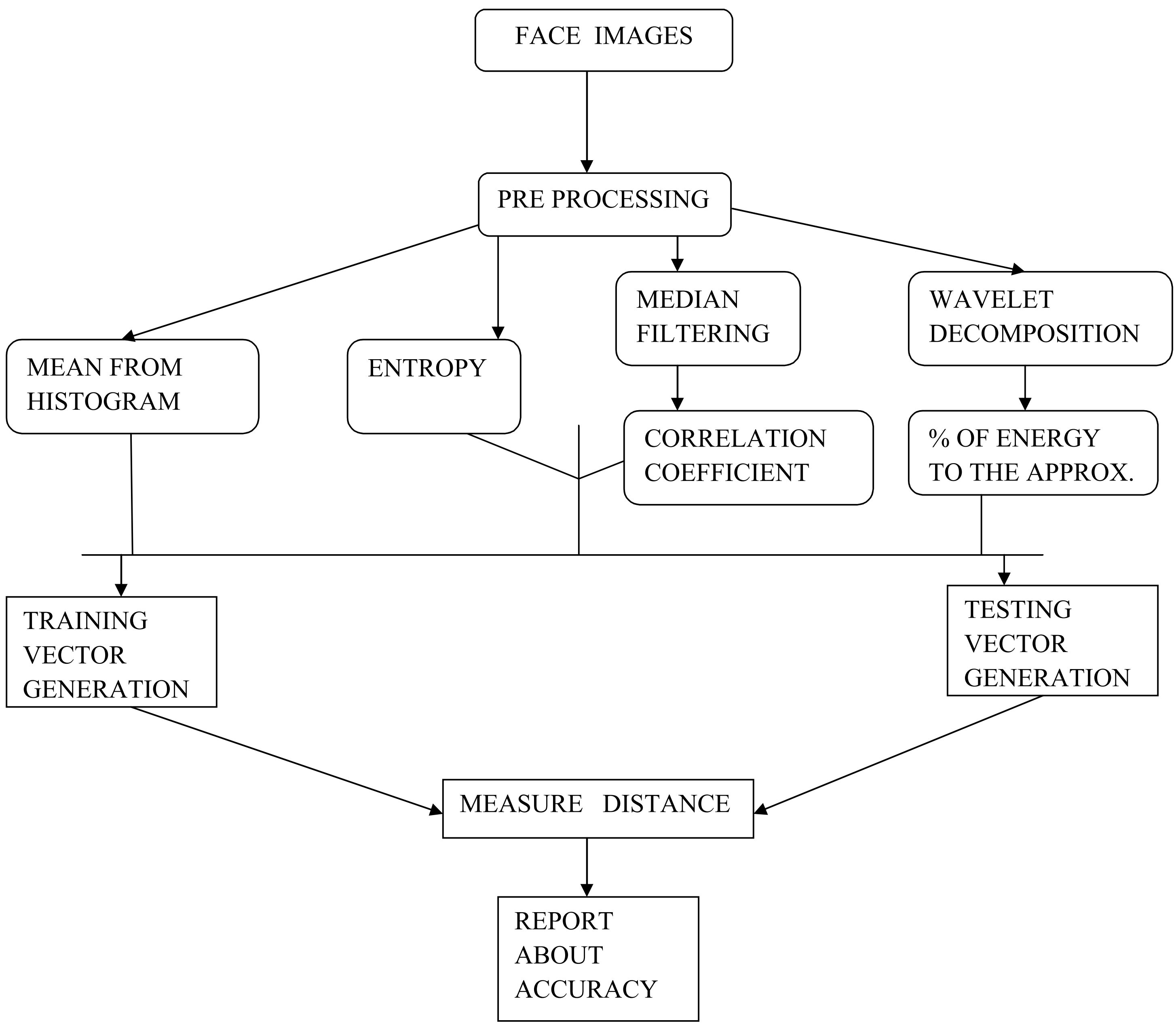


Fig 1: Flow diagram of the entire process

In this paper we propose face recognition using wavelet decomposition and mean from histogram.

Our first approach on pre processing is to convert the image from RGB to gray scale . For this we have used some functions that converts the true color image RGB to to the grayscale intensity image.

Our second approach on pre processed image is to calculate the value of mean of histogram, entropy, median filtering and wavelet decomposition. For this we have used some functions to get the values. After median filtering we have used some functions to get the value of correlation coefficient. And after wavelet decomposition we have used a function to get the value (in percentage) of energy corresponding to the approximation.

Our third approach is to generate the feature vector by using these values. We have used entropy, mean of histogram, correlation coefficient, and percentage of energy as a combination of feature vector . So, if the feature vector is E den E can be represented as

E=[entropy ,mean of histogram, correlation coefficient, % of energy]

So we have calculated the feature vectors for all the training and testing images .To get the average value we have taken into consideration the mean value of each training class.

Our last approach is to check the difference between the test image feature vector to the each training class's mean value and to find out the minimum difference.

3.2 FEATURE VECTOR GENERATION

3.2.1 Preprocessing:

According to [23], Humans perceive color through wavelength-sensitive sensory cells called cones. There are three different types of cones, each with a different sensitivity to electromagnetic radiation (light) of different wavelength. One type of cone is mainly sensitive to red light, one to green light, and one to blue light. By emitting a controlled combination of these three basic colors (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any perceivable color. This is the reasoning behind why color images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such color images as stored in an RGB format.

In grayscale images, however, we do not differentiate how much we emit of the different colors, we emit the same amount in each channel. What we can differentiate is the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels.
When converting an RGB image to grayscale, we have to take the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One such approach is to take the average of the contribution from each channel: (R+B+C)/3. However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to take a weighted average, e.g.: 0.3R + 0.59G + 0.11B.
A different approach is to let the weights in our averaging be dependent on the actual image that we want to convert, i.e., be adaptive. A (somewhat) simple take on this is to form the weights so that the resulting image has pixels that have the most variance, since pixel variance is linked to the contrast of the image. In the applet above, the "optimal projection" calculates how we should combine the RGB channels in the selected image to make a grayscale image that has the most variance. [For the more technically advanced; we find the weights by taking the principal eigenvector of the sample covariance matrix of the RGB channels.]
We convert RGB image or colormap to grayscale. We convert the true color image RGB to the grayscale intensity image . We convert RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. If I is an RGB image, it can be of class uint8, uint16, single, or double. The output image I is of the same class as the input image. If I is a color map, the input and output color maps are both of class double.

3.2.2 Histogram:

According to [17], the histogram of a monochrome image is a graphical representation of the frequency of occurrence of each gray level in the image. The data structure that stores the Frequency values is a 1D array of numerical values, h, whose individual elements store the number(or percentage)of image pixels that correspond to each possible gray level. Each individual histogram entry can be expressed mathematically as

$$h(k)=nk=card\{(x,y)|f(x,y)=k\} \qquad (1)$$ [source[17]]

Here, k=0,1, ...,L−1,where L is the number of gray level so f the digitized image ,and card{···}denotes the cardinality of a set, that is, the number of elements in that set(nk).
A normalized histogram can be mathematically defined as

$$p(rk)=nk/n \qquad (2)$$ [source[17]]

Where n is the total number of pixels in the image and p(rk) is the probability(percentage) of the kth gray level(rk). Histograms are normally represented using a bar chart, with one bar per gray level, In which the height of the bar is proportional to the number (or percentage) of pixels that correspond to that particular gray level.

3.2.3 Extract mean from histogram:

Estimate the mean by using the bin counts and the bin widths and insert in the integral formula for the mean. The estimate will be more or less good, depending on the distribution of samples inside each bin interval.

3.2.4 Entropy:

According to [21], Entropy is a concept which originally arose from the study of the physics of heat engines. It can be described as a measure of the amount of disorder in a system. An organized structure, such as a crystal or a living organism, is very highly ordered and consequently has low entropy. When the crystal is heated sufficiently, it melts and becomes liquid, a much less ordered state. When the organism dies, it decays and becomes completely disrupted. In either system, its entropy increases.

Another way of expressing entropy is to consider the spread of states which a system can adopt. A low entropy system occupies a small number of such states, while a high entropy system occupies a large number of states.

In the case of an image, these states correspond to the gray levels which the individual pixels can adopt. For example, in an 8-bit pixel there are 256 such states. If all such states are equally occupied, as they are in the case of an image which has been perfectly histogram equalized, the spread of states is a maximum, as is the entropy of the image. On the other hand, if the image has been threshold, so that only two states are occupied, the entropy is low. If all of the pixels have the same value, the entropy of the image is zero.

Note that, in this progression, as the entropy of the image is decreased, so is its information content. We moved from a full gray scale image, with high entropy, to a threshold binary image, with low entropy, to a single-valued image, with zero entropy.

Information, in this context, refers to the announcement of the unexpected. If the pixels of an image were inspected, and found to be the same. This information could have been communicated in a very short message. The information content is said to be low simply because it can be communicated in a short message. If the pixels are changing in unexpected ways, however, longer messages are required to communicate this fact and the information is said to increase. This assumes, of course, that all changes in the image are meaningful.

Changes due to noise are still considered to be information in that they describe the image as it actually is, rather than as it should be.

Noise and heat play similar roles in increasing the entropy of systems. The entropy H of an image is defined as

$$H = -\sum_{k=0}^{M-1} p_k \log_2(p_k)$$

(3)[Source [21]]

Where M is the number of gray levels and p_k is the probability associated with gray level k. Maximum entropy is achieved in the case of a uniform probability distribution. If $M = 2^n$, then p_k is constant and given by

$$p_k = \frac{1}{M} = 2^{-n}$$

(4) [Source [21]]

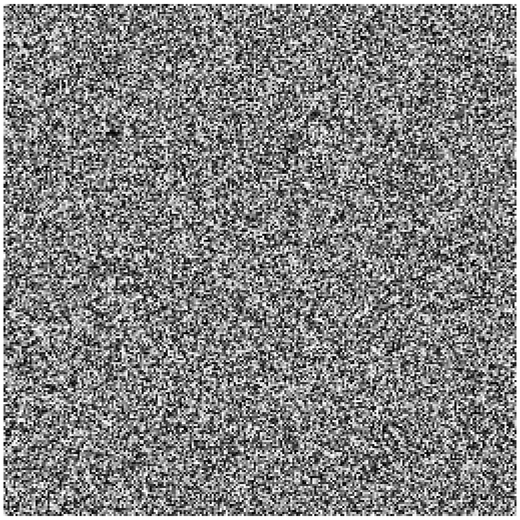
The maximum entropy is found from

$$H_{\max} = -\sum_{k=0}^{M-1} (1/M) \log_2(1/M) = -\log(2^{-n}) = n$$

(5)[Source [21]]

Minimum entropy is achieved when the image itself is constant, that is, all of the pixels have the same gray level k. For that gray level, $p_k = 1$, and H = -log(1) = 0.

Entropy sets a lower bound on the average number of bits per pixel required to encode an image without distortion. This is only true, however, for uncorrelated images. Consider the images below, [Source [21]]



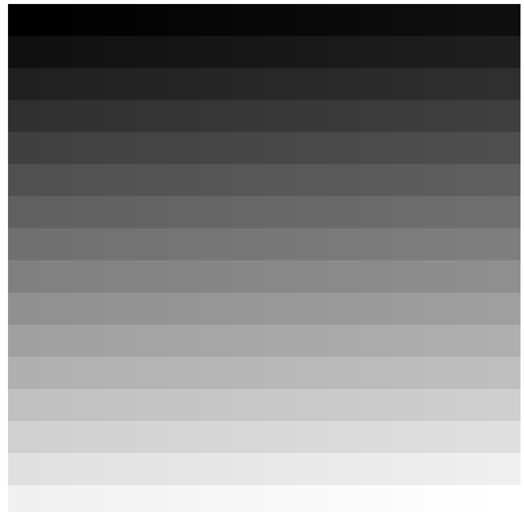


Fig2a:Image with uniform random noise Fig2b:Image with distribution of gray levels

The image on the left (uniform random noise) has an entropy of 8 bits and is uncompressible .The image on the right has the same distribution of gray levels but is highly spatially correlated.

3.2.5 Double:

It converts the intensity image to double precision, rescaling the data if necessary and returns the double-precision value for X where X is an image matrix. If X is already a double-precision array, double has no effect.

3.2.6 Median Filtering:

According to [20], Median filtering is a nonlinear process useful in reducing impulsive or salt-and-pepper noise. It is also useful in preserving edges in an image while reducing random noise. Impulsive or salt-and pepper noise can occur due to a random bit error in a communication channel. In a median filter, a window slides along the image, and the median intensity value of the pixels within the window becomes the output intensity of the pixel being processed. For example, suppose the pixel values within a window are 5,6, 55, 10 and 15, and the pixel being processed has a value of 55. The output of the median filter an the current pixel location is 10, which is the median of the five values.

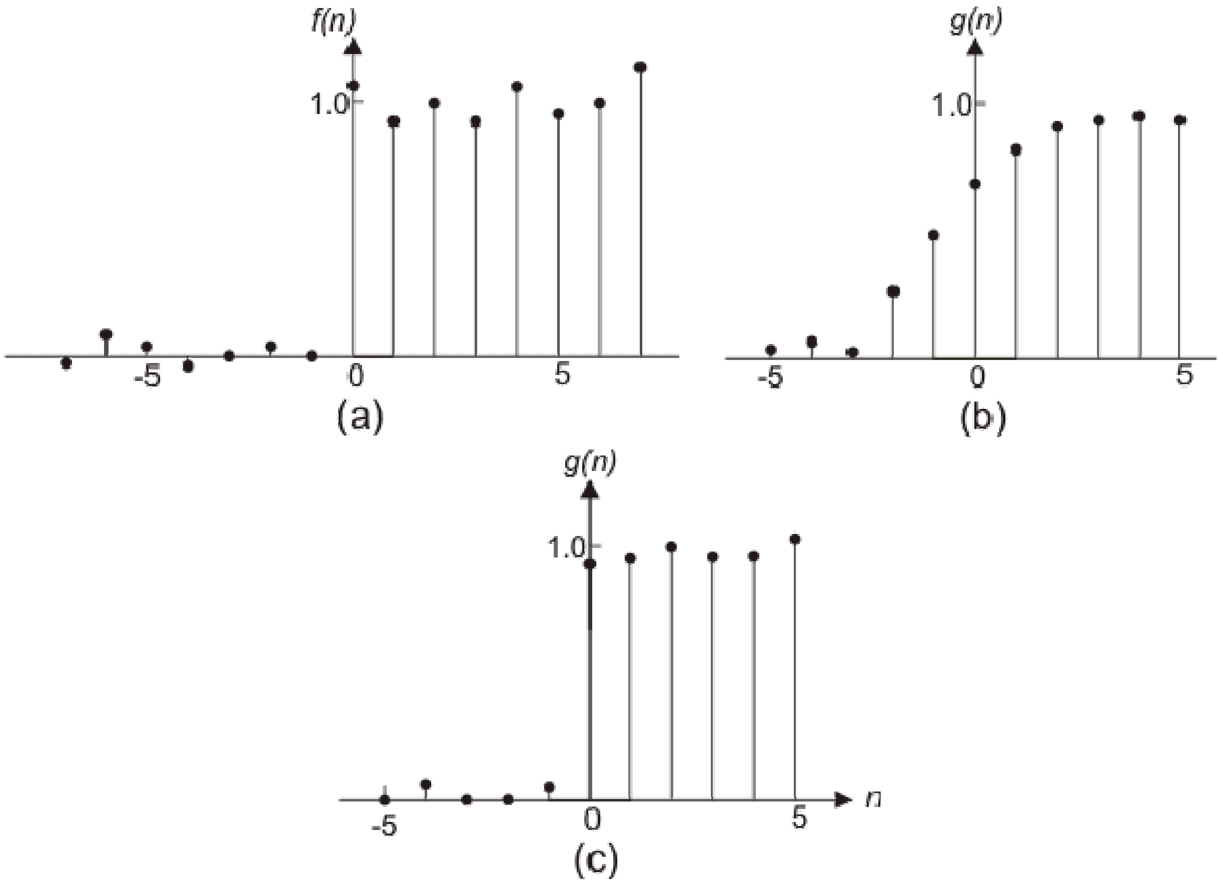


Fig3: 1-D step sequence degraded by a small amount of random noise. [source[20]]

Like lowpass filtering, median filtering smoothes the image and is thus useful in reducing noise. Unlike lowpass filtering, median filtering can preserve discontinuities in a step function and can smooth a few pixels whose values differ significantly from their surroundings without affecting the other pixels.Figure (3) shows a 1-D step sequence degraded by a small amount of random noise. Figure (3) shows the result after filtering with a lowpass filter whose impulse response is a 5-point rectangular window. Figure (3) shows the result after filtering with 5-point median filter. It is clear from the figure that the step discontinuity is better preserved by the median filter. Figure (3a) shows a 1-D sequence with two values that are significantly different from the surrounding points. Figures (b) and (c) show the result of a lowpass filter and a median filter, respectively. The filters used in figure (4) are the same as those used in figure(3). If the two impulsive values are due to noise, the result of using a median filter will be the reduce the noise. If the two values are part of the signal, however, using the median filter will distort the signal.

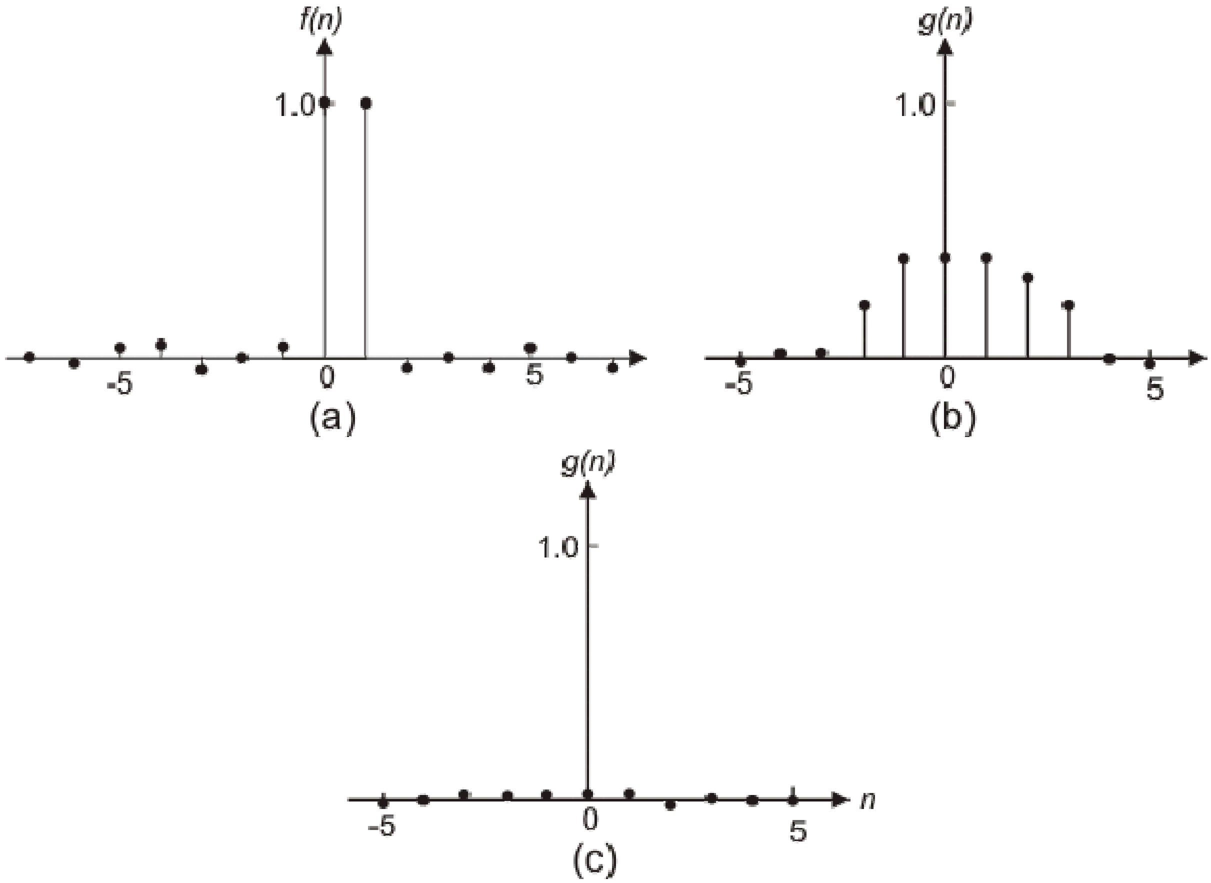


Fig4: filtration of two impulsive values[source [20]]

The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.)

An important parameter in using a median filter is the size of the window. Figure (5) illustrates the result of median filtering the signal in Figure 4 (a) as a function of the window size. If the window size is less than 5, the two pixels with impulsive values will not be significantly affected. For a large window, they will be. Thus, the choice of the window size depends on the context. Because it is difficult to choose the optimum window size in advance, it may be useful to try several median filters of different window sizes and choose the best of the resulting images.

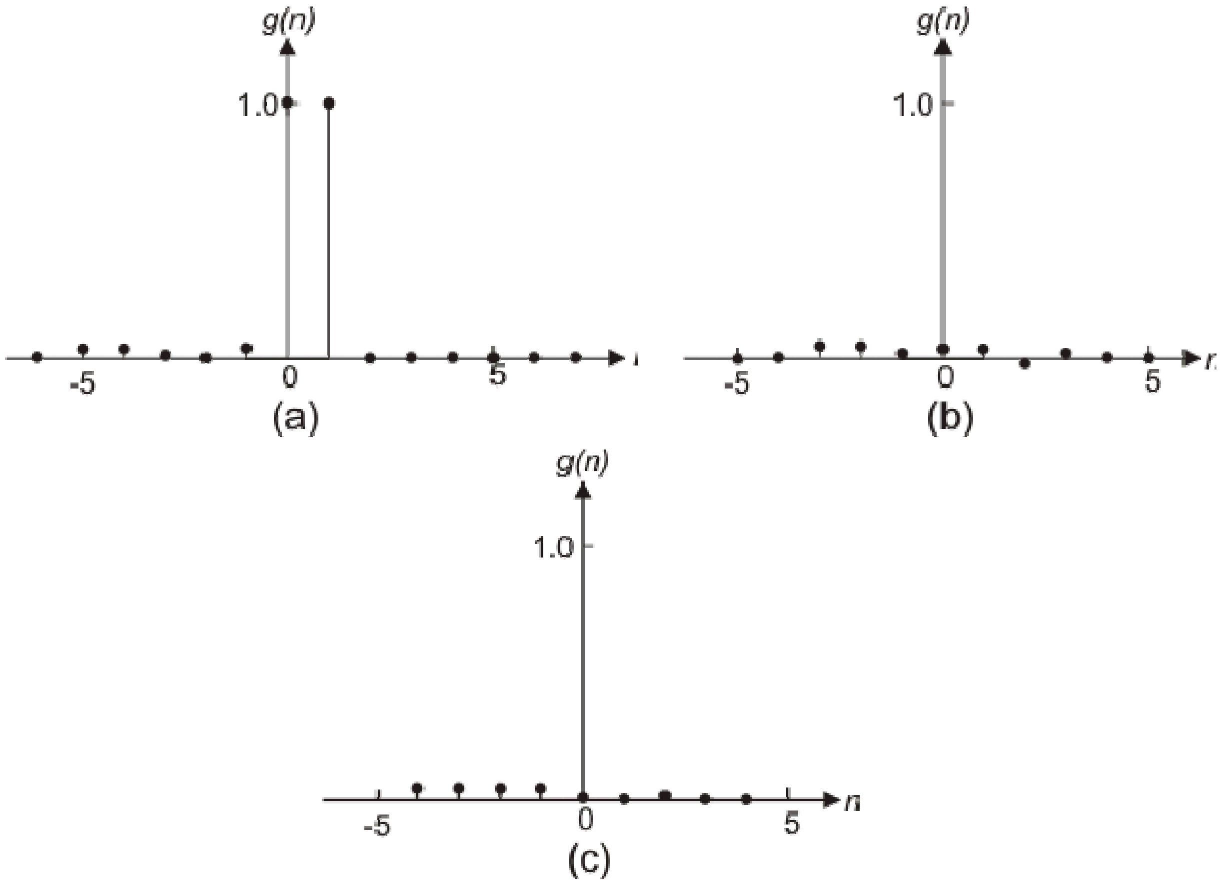


Fig5: result of median filtering [source [20]]

In the above, we discussed I-D median filtering. The task involved in performing a median filtering operation extends straightforwardly from the 1-D case to the 2-D case. However, not all properties of a 1-D median filter apply to a 2-D median filter. For example, median filtering a 1-D unit seep sequence u(n) preserves the step discontinuity and does not affect the signal u(n) at all. Suppose we filter a 2-D step sequence $u(n_1, n_2)$ with a 2-D N X N-point median filter. Figure (6a) shows $u(n_1,n_2)$ and Figure(6b) shows the result of filtering $u(n_1, n_2)$ with a 2-D 5 x 5–point median filter. Suppose we filter a 2-D step sequence $u(n_1, n_2)$ with a 2-D N X N-point median filter. From Figure (b), the intensity discontinuities which can be viewed as 1-D steps (for large n_1 and $n_2 = 0$ and large n_2 at $n_1 = 0$) are seriously distorted. One method that tends to preserve 2-D step discontinuities will is to filter a 2-D signal along the horizontal direction with a 1-D median filter and then filter the result along the vertical direction with another 1-D median filter. This method is called separable median filtering, and is often used in 2-D median filtering applications. When a separable median filter is applied to $u(n_1, n_2)$ the signal $u(n_1, n_2)$is not affected.

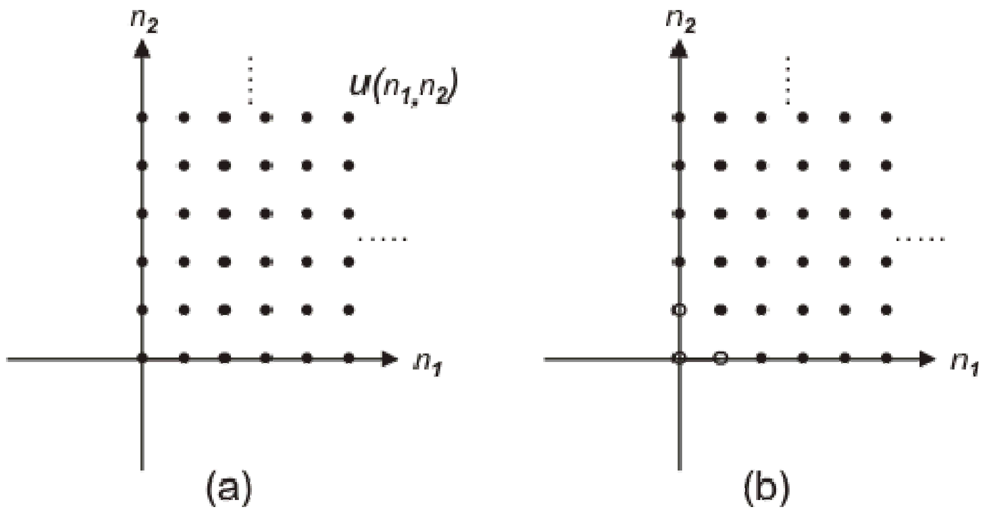


Fig6: 2-D 5 x 5–point median filter. [source [20]]

A median filter is a nonlinear system, and therefore many theoretical results on linear systems are not applicable. For example, the result of separable median filtering depends on the order in which the 1-D horizontal and vertical median filters are applied. Despite this difficulty, some theoretical results have been developed on median filtering. One result states that repeated application of a 1-D median filter to a 1-D sequence eventually leads to a signal called a "root signal" which is invariant under further applications of the 1-D median filter.

The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.

3.2.7 Correlation Coefficient:

According to Eugene and Johnston in [11], the Pearson's Correlation Coefficient, r, is widely used in statistical analysis, pattern recognition, and image processing. Applications on the latter include comparing two images for image registration purposes, object recognition, and disparity measurement. For monochrome digital images, the Pearson's Correlation Coefficient is described by Eq(6):

$$r = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\sqrt{\sum_i (y_i - y_m)^2}}$$

(6)[source [11]]

where xi is the intensity of the ith pixel in image 1, yi is the intensity of the ith pixel in image 2, xm is the mean intensity of image 1, and ym is the mean intensity of image 2. The correlation coefficient has value 1 if the two images are identical, 0 if they are completely uncorrelated, and –1 if they are completely anti-correlated, for example, if one image is the negative of the other. In theory, they would obtain a value of 1 for r if the object is intact and a value of less than 1 if alteration or movement has occurred. In practice, distortions in the imaging system, pixel noise, slight variations in the object's position relative to the camera, and other factors produce an r value less than 1, even if the object has not been moved or physically altered in any manner.

3.2.8 Wavelet Decomposition:

According to [12], we often want to get an images' multi-stage decomposition for a small wave, so that we can have a more accurate analysis of wavelet.

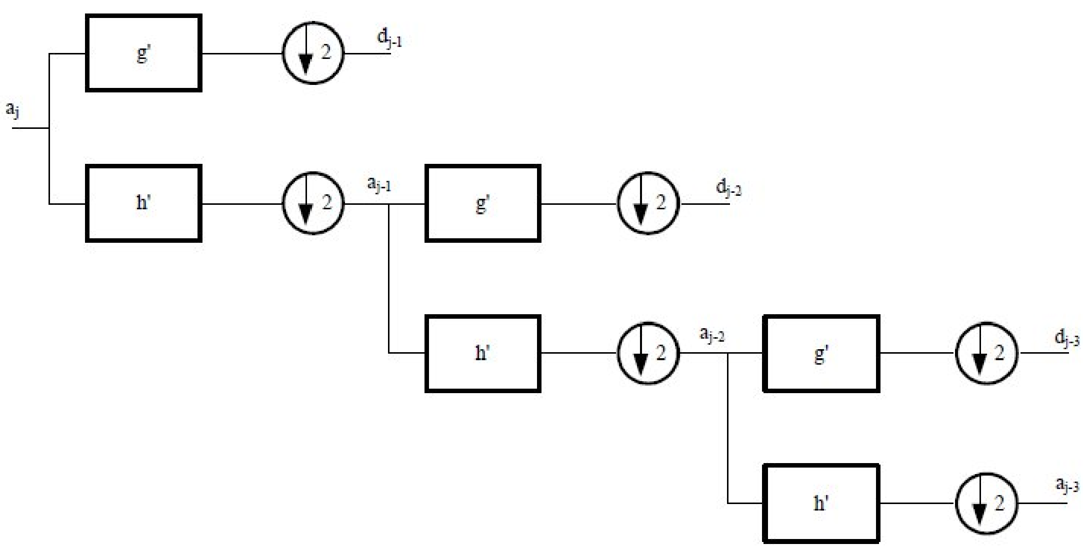


Fig7: wavelet decomposition [source [12]]

In figure 7, 'h' is low-pass filter, 'g' is high-pass filter, '↓2 'is down sampling.

From the above we can see multistage of wavelet decomposition process clearly from the schematic diagram, but in the process of research, we need to abstract into specific formulas, so that we will use more widely. Hence, we translate the process. For a given signal sampling, first approximate f and fj with f, through the decomposition theorem, it is decomposed ck into fj and dk

Set[1]

$$f_j(x) = \sum_{k \in z} C_k^j \varphi(2^j x - k) \in v_j \quad (1)$$

f_j can be broken down into $f_j = w_{j-1} + f_{j-1}$,in which

,

$$w_{j-1} = \sum_{k \in z} d_k^{j-1} \psi(2^{j-1} x - k) \in w_{j-1} \quad (2)$$

$$f_{j-1} = \sum_{k \in z} C_k^{j-1} \varphi(2^{j-1} x - k) \in v_{j-1} \quad (3)$$

Fig8: wavelet decomposition formulas [source [12]]

3.2.9 Wavelet Energy:

Energy in image processing has different meaning depending on the context. There are more than one definitions of energy in image processing so it depends on the context of where it was used. It depends on the context, but in general, in Signal Processing, "energy" corresponds to the mean squared value of the signal (typically measured with respect to the global mean value). This concept is usually associated with the Perceval theorem, which allows us to think of the total frequency as distributed along coordinates of tile i in the goal state, the heuristic is:

$$h(s) - \sum_{i=1}^{0} (|x_i(s) - \bar{x}_i| + |y_i(s) - \bar{y}_i|) \qquad (7)$$

Wavelet energy is denoted by Ea, which is the percentage of energy corresponding to the approximation and Ed, which is the vector containing the percentages of energy corresponding to the details. The vector which contains the percentages of energy corresponding to the terminal nodes of the tree T is denoted by E . In this case, wenergy is a method of the wptree object T

3.2.10 Feature Vector:

According to [22], in computer vision and image processing, a feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image.

Other examples of features are related to motion in image sequences, to shapes defined in terms of curves or boundaries between different image regions, or to properties of such a region.

The feature concept is very general and the choice of features in a particular computer vision system may be highly dependent on the specific problem at hand.When features are defined in terms of local neighborhood operations applied to an image, a procedure commonly referred to as feature extraction, one can distinguish between feature detection approaches that produce local decisions whether there is a feature of a given type at a given image point or not, and those who produce non-binary data as result. The distinction becomes relevant when the resulting detected features are relatively sparse. Although local decisions are made, the output from a feature detection step does not need to be a binary image. The result is often represented in terms sets of (connected or unconnected) coordinates of the image points where features have been detected, sometimes with sub pixel accuracy.

When feature extraction is done without local decision making, the result is often referred to as a feature image. Consequently, a feature image can be seen as an image in the sense that it is a function of the same spatial (or temporal) variables as the original image, but where the pixel values hold information about image features instead of intensity or color. This means that a feature image can be processed in a similar way as an ordinary image generated by an image sensor. Feature images are also often computed as integrated step in algorithms for feature detection.A specific image feature, defined in terms of a specific structure in the image data, can often be represented in different ways. For example, an edge can be represented as a boolean variable in each image point that describes whether an edge is present at that point. Alternatively, we can instead use a representation which provides a certainty measure instead of a boolean statement of the edge's existence and combine this with information about the orientation of the edge. Similarly, the color of a specific region can either be represented in terms of the average color (three scalars) or a color histogram (three functions).

When a computer vision system or computer vision algorithm is designed the choice of feature representation can be a critical issue. In some cases, a higher level of detail in the description of a feature may be necessary for solving the problem, but this comes at the cost of having to deal with more data and more demanding processing. Below, some of the factors which are relevant for choosing a suitable representation are discussed. In this discussion, an instance of a feature representation is referred to as a (feature) descriptor.
In some applications it is not sufficient to extract only one type of feature to obtain the relevant information from the image data. Instead two or more different features are extracted, resulting in two or more feature descriptors at each image point. A common practice is to organize the information provided by all these descriptors as the elements of one single vector, commonly referred to as a feature vector. The set of all possible feature vectors constitutes a feature space.

A common example of feature vectors appears when each image point is to be classified as belonging to a specific class. Assuming that each image point has a corresponding feature vector based on a suitable set of features, meaning that each class is well separated in the corresponding feature space, the classification of each image point can be done using standard classification method.

Another, and related example, occurs when neural network based processing is applied to images. The input data fed to the neural network is often given in terms of a feature vector from each image point, where the vector is constructed from several different features extracted from the image data. During a learning phase, the networks can itself find which combinations of different features that are useful for solving the problem at hand.

3.3 ALGORITHM

STEP 1: capture and create the dataset and divide the dataset into two parts. One for training phase and another for testing phase.

STEP 2: start with the training images of each class and continue till all classes' training data is processed.

STEP 3: read the RGB image.

STEP 4: convert the image into grayscale.

STEP 5: apply histogram function and store the values.

STEP 6: find the mean from histogram.

STEP 7: find the entropy and store the value.

STEP 8: change the array into double precision array.

STEP 9: apply median filter and store the value.

STEP 10: find the correlation coefficient and store the value.

STEP 11: use wavelet decomposition onto the image and store the values.

STEP 12: find out the percentage of energy corresponding to the details and store the value.

STEP 13: use the feature vector as the combination of output of STEP 6,7,10 and 12

STEP 14: find the mean of each classes' training data and store the value.

STEP 15: start reading images from testing phase.

STEP 16: continue STEP 4 to STEP 13.

STEP 17: calculate the difference between the feature vectors of particular test image to the each output of STEP 14.

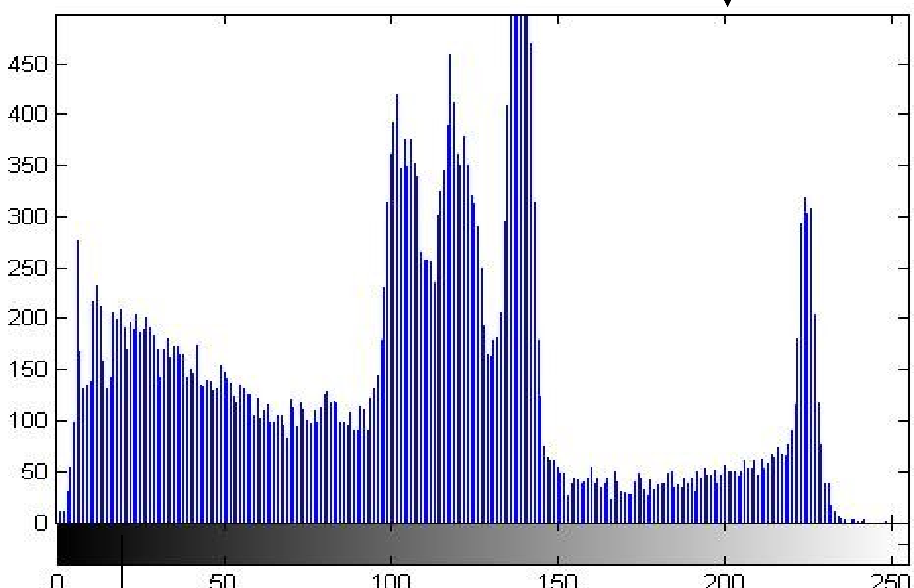
STEP 18: find the minimum difference and calculate accuracy.

STEP 19: END

3.4 EXPERIMENTS

rgb2gray

Histogram



Find the mean from histogram, here meanGL=106.5257

Entropy, here e=7.3950

Double (I)

Median Filter

Median Filter ⟶ correlation coefficient, here R=0.9988

↓

Wavelet decomposition

↓

Wavelet energy, here Ea=96.4390

If E is a feature vector then E is denoted by

E= [meanGL	R	E	EA]
= [106.5257	0.9988	7.3950	96.4390]

3.5 MANHATTAN DISTANCE

Similarity measures play important role in retrieval systems. Manhattan is a very commonly use distance measurement, which can be used to measure feature closeness .The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components and obtained in following equation. In this study, images are represented by a feature vector, thus Manhattan distance classifier is used to measure the similarity values with comparisons between training face images and testing face images for computing the recognition rates.

$$MD=\sum \left| Si - Ti \right| \qquad (7)$$

where i lies between 1 to n Where n is a number of feature and MD is Manhattan Distance between the testing face images Si and the training face images Ti.

CHAPTER 4

EXPERIMENTATIONS AND RESULTS

4.1 *DATASET*

For experimentations of face recognition face dataset from Computer Vision Science Research Projects [18] is used. We have worked with 50 classes of that dataset and we have chosen 7 images of each class for my algorithm. The images are named in numeric from 1(1) to 1(7) where 1 is the class number and number inside the brackets specifies the image number. We have used 4 images of each class as train image and the rest 3 images for the test purpose.

No Of Class : 50

No Of Train Image For Each Class : 4
No Of Test Iamge For Each Class : 3
Total Image : 50*7=350
Image Extension : ".jpg"
Here we have given some data from the dataset we have worked on.

Fig 9: DATASET

4.2 *MEAN OF EACH CLASS'S TRAINING IMAGES*

Matrix M is calculated the mean of feature values of the training face images. In the M matrix the row indicates the number training classes and column indicates the number of training samples. After calculating the M matrix contains 50 rows and 4columns

Here, we are giving the mean of 50 classes' training vector.

TABLE 1: MEAN OF EACH CLASS'S TRAINING IMAGE

Mean of training vector(class no)	correlation coefficient	Wavelet energy	Entropy	mean from histogram
1	0.9990	97.4657	7.3934	99.8820
2	0.9989	96.4350	7.3913	107.0136
3	0.9989	96.9371	7.6361	115.9136
4	0.9979	97.0243	6.8494	76.0887
5	0.9986	97.3582	7.1681	91.9357
6	0.9985	97.0060	7.2330	107.8086
7	0.9983	97.0308	7.4026	105.9643
8	0.9969	97.9710	7.3357	120.0787
9	0.9986	96.6611	7.5701	104.4067
10	0.9994	96.3278	7.3314	77.9738
11	0.9975	96.3250	7.4010	108.8215
12	0.9983	97.4239	7.4168	118.5840
13	0.9980	97.8585	7.2145	126.8146
14	0.9969	97.0115	7.2664	93.5784
15	0.9972	97.6059	7.3397	128.4905
16	0.9987	96.4462	7.7054	114.9509
17	0.9989	97.3524	7.3478	103.2788

18	0.9982	97.1351	7.5267	123.3815
19	0.9993	96.6641	7.2151	103.7022
20	0.9985	97.8360	7.3821	110.6834
21	0.9977	96.6686	6.0435	47.1932
22	0.9819	92.3087	6.5163	56.9717
23	0.9785	94.1285	6.3574	47.1948
24	0.9932	97.4785	5.9154	51.4122
25	0.9974	97.8831	5.7355	56.6552
26	0.9914	96.5525	5.9102	36.5201
27	0.9894	94.2396	6.2482	53.6164
28	0.9802	97.1693	5.8813	59.4632
29	0.9946	95.5899	5.6625	39.2084
30	0.9950	95.3121	7.0601	80.9504
31	0.9973	97.7959	6.7116	90.0971
32	0.9746	96.4268	6.4392	67.8863
33	0.9967	95.0103	5.9108	59.5347
34	0.9988	97.5733	7.1880	98.1787
35	0.9986	97.1085	7.3999	82.1886
36	0.9978	96.5374	7.4525	97.2427
37	0.9995	97.6944	7.5173	87.1629
38	0.9993	96.1757	6.5153	48.6443
39	0.9992	96.1271	7.3908	83.0305
40	0.9981	93.3985	7.0999	73.5697
41	0.9982	96.3841	6.5673	51.0973

42	0.9972	94.4531	6.4942	56.4043
43	0.9990	95.7595	7.3961	81.7409
44	0.9991	93.8350	7.1026	69.7189
45	0.9985	95.5080	7.4704	83.5682
46	0.9982	96.7665	6.3204	73.1214
47	0.9979	96.5654	5.6106	28.8619
48	0.9960	97.2916	6.4674	77.4000
49	0.9981	97.5439	7.4956	116.1166
50	0.9856	97.4547	7.4021	124.9991

4.2.1*Plot of Mean Feature Value of Correlation coefficient of each Training Set*

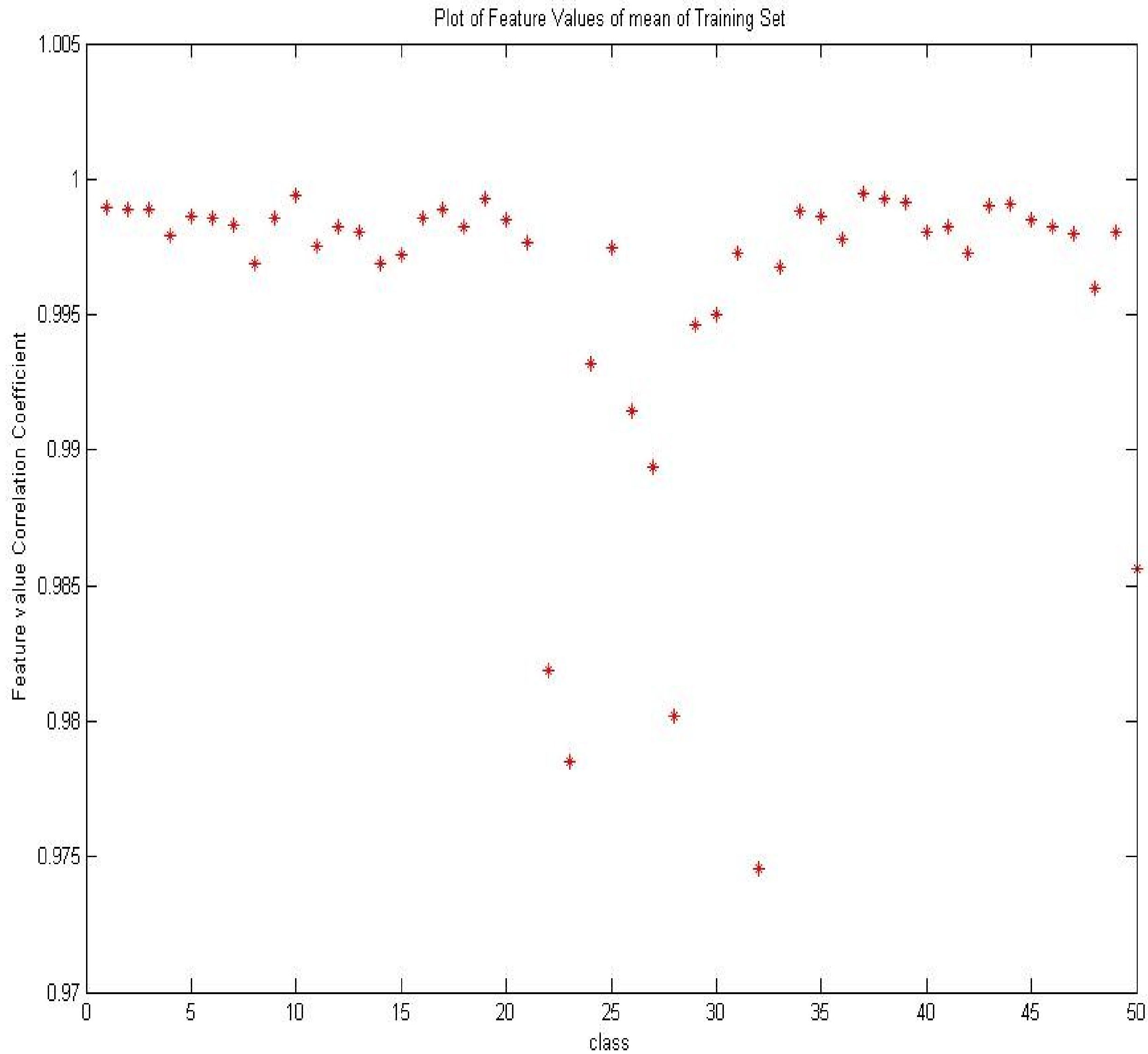


Fig 10 : Plot of R Feature Values of Training Set

4.2.2 *Plot of Mean Feature Value of % of energy of each Training Set*

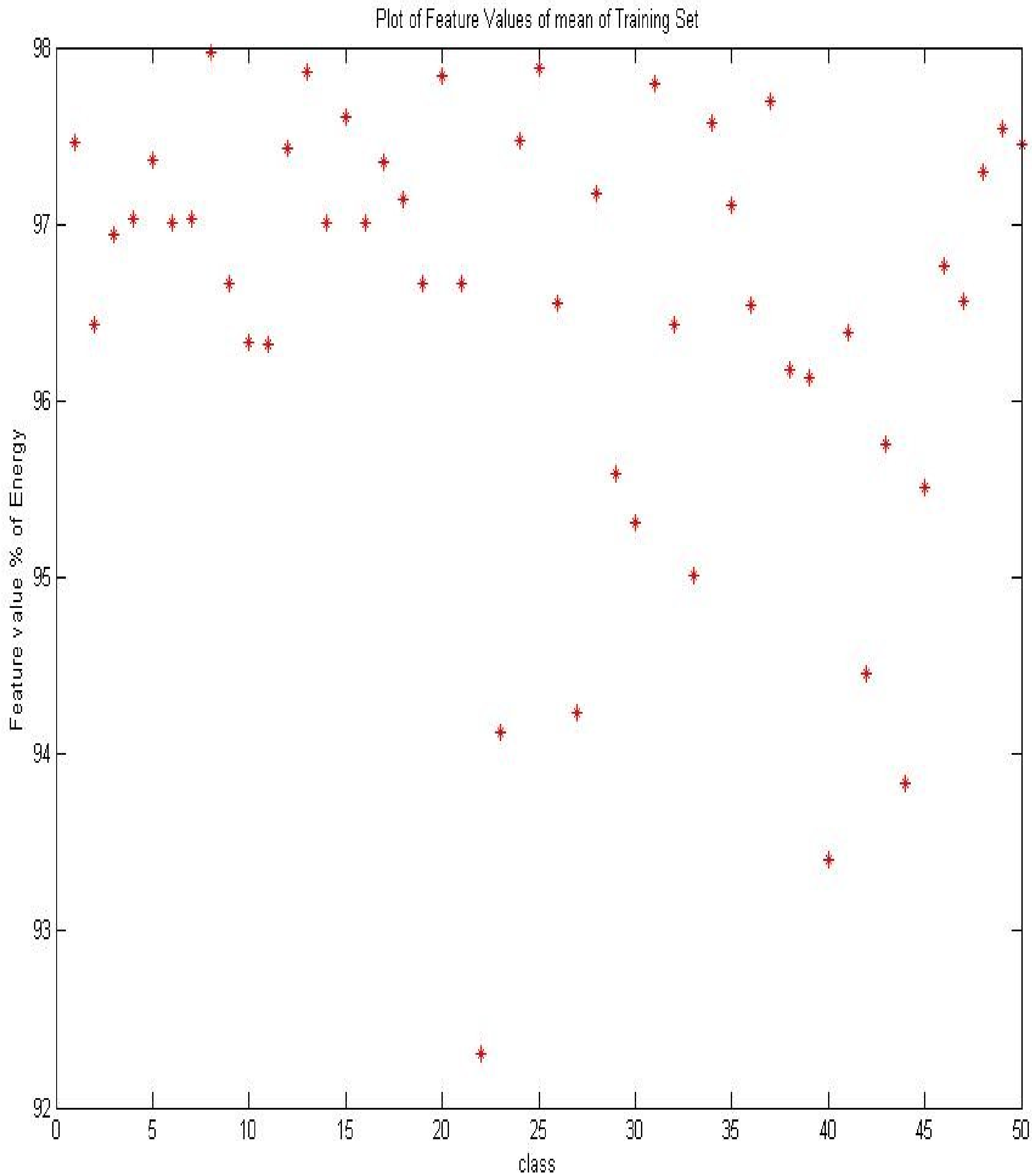


Fig 11 : Plot of Ea Feature Values of Training Set

4.2.3 *Plot of Mean Feature Value of Entropy of each Training Set*

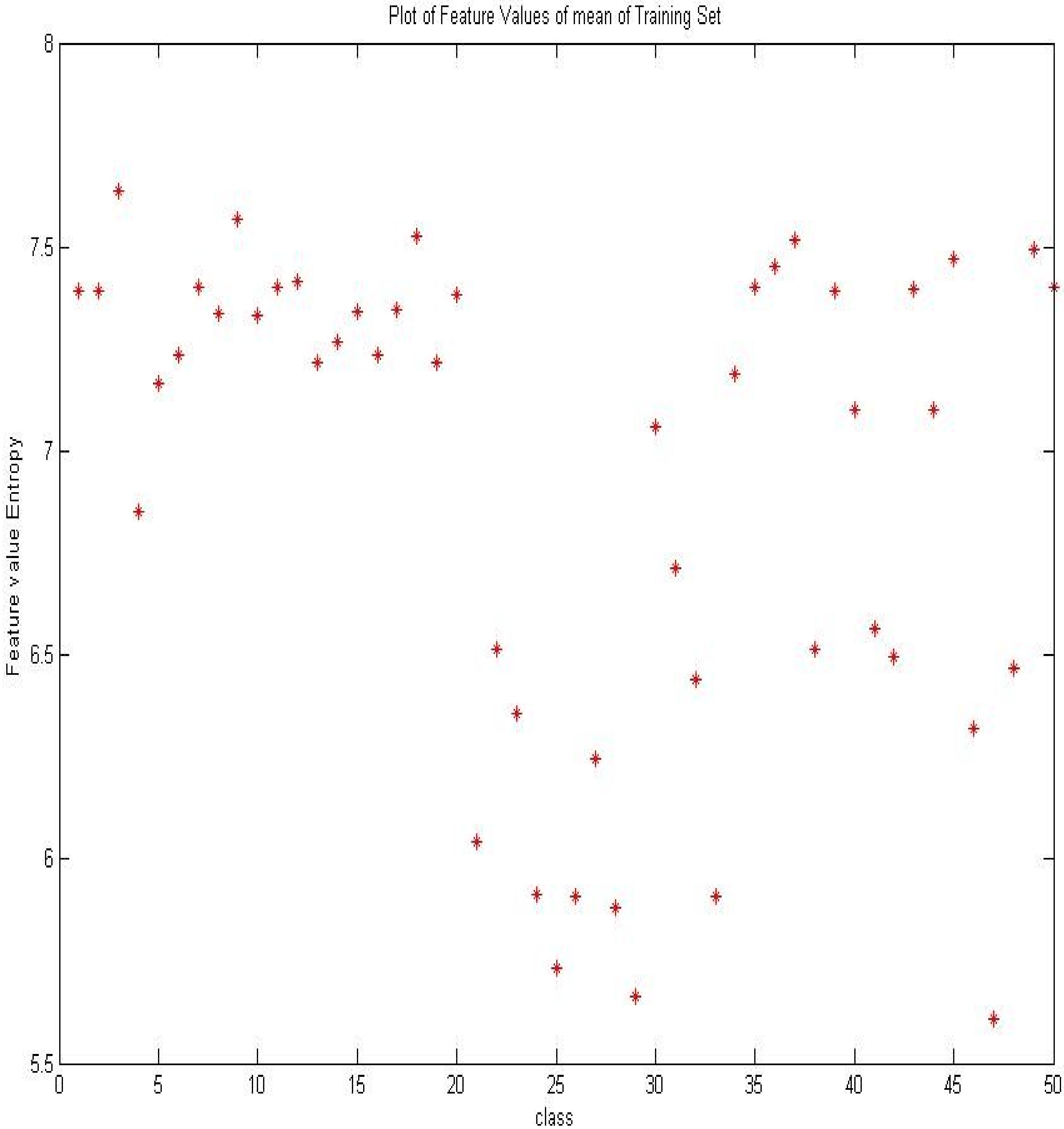


Fig 12 : Plot of e feature Values of Training Set

4.2.4 *Plot of Mean Feature Value of mean from histogram of each Training Set*

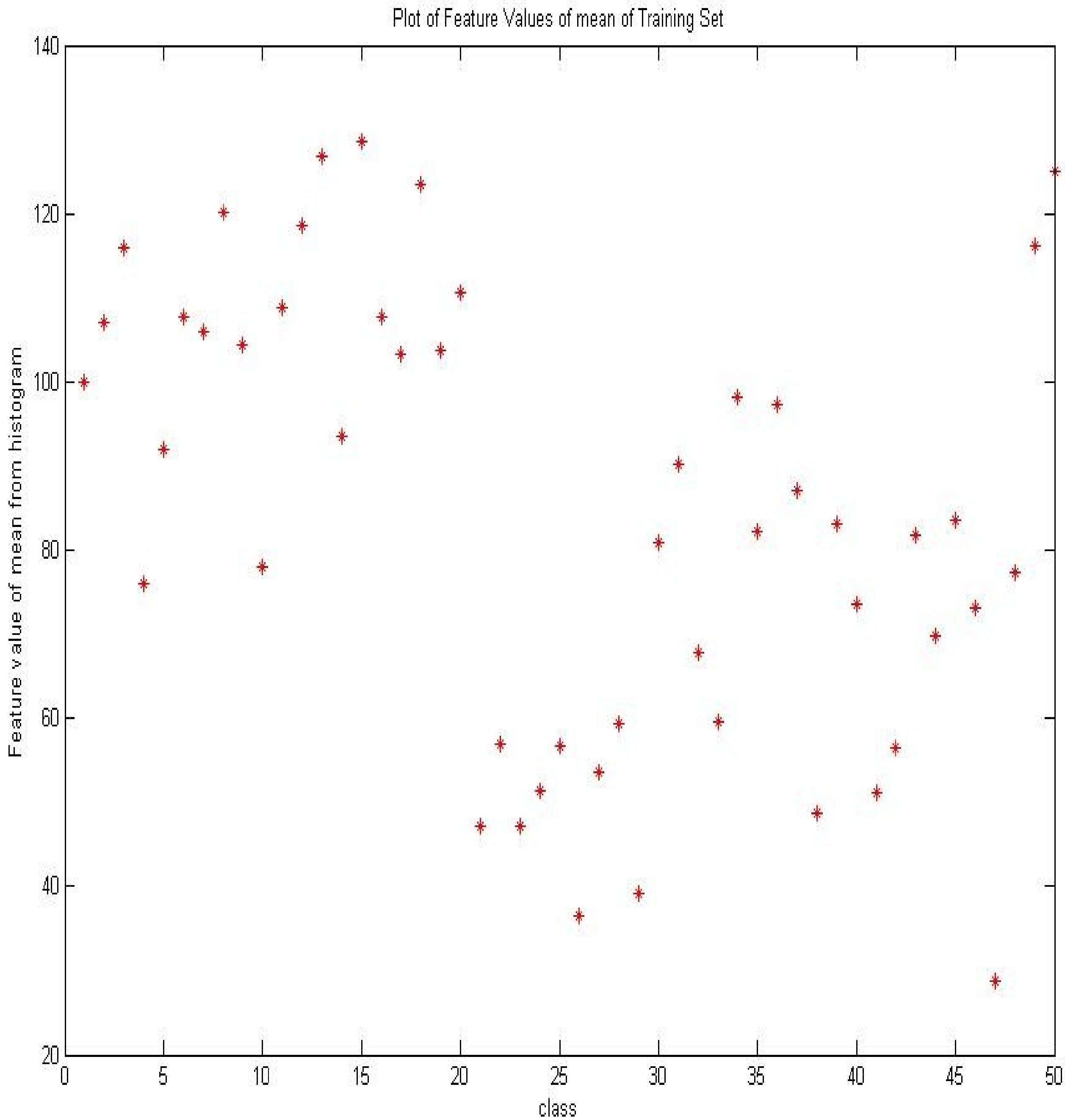


Fig 13 : Plot of mean from histogram feature Values of Training Set

4.3 *FEATURE VECTOR OF TEST IMAGES*

Here we have taken a set of testing face images of 1 class. S is the test vector.

Here we are giving 3 sample test data of class 1.

TABLE 2: FEATURE VECTOR OF TEST IMAGES

Test image Data Class No	correlation coefficient	Wavelet energy	Entropy	mean from histogram
1	0.9990	97.4659	7.3848	99.7365
1	0.9990	97.5112	7.3924	100.0929
1	0.9990	97.5032	7.3778	99.7630

4.3.1 *Plot of Feature Value of correlation coefficient of Testing Set*

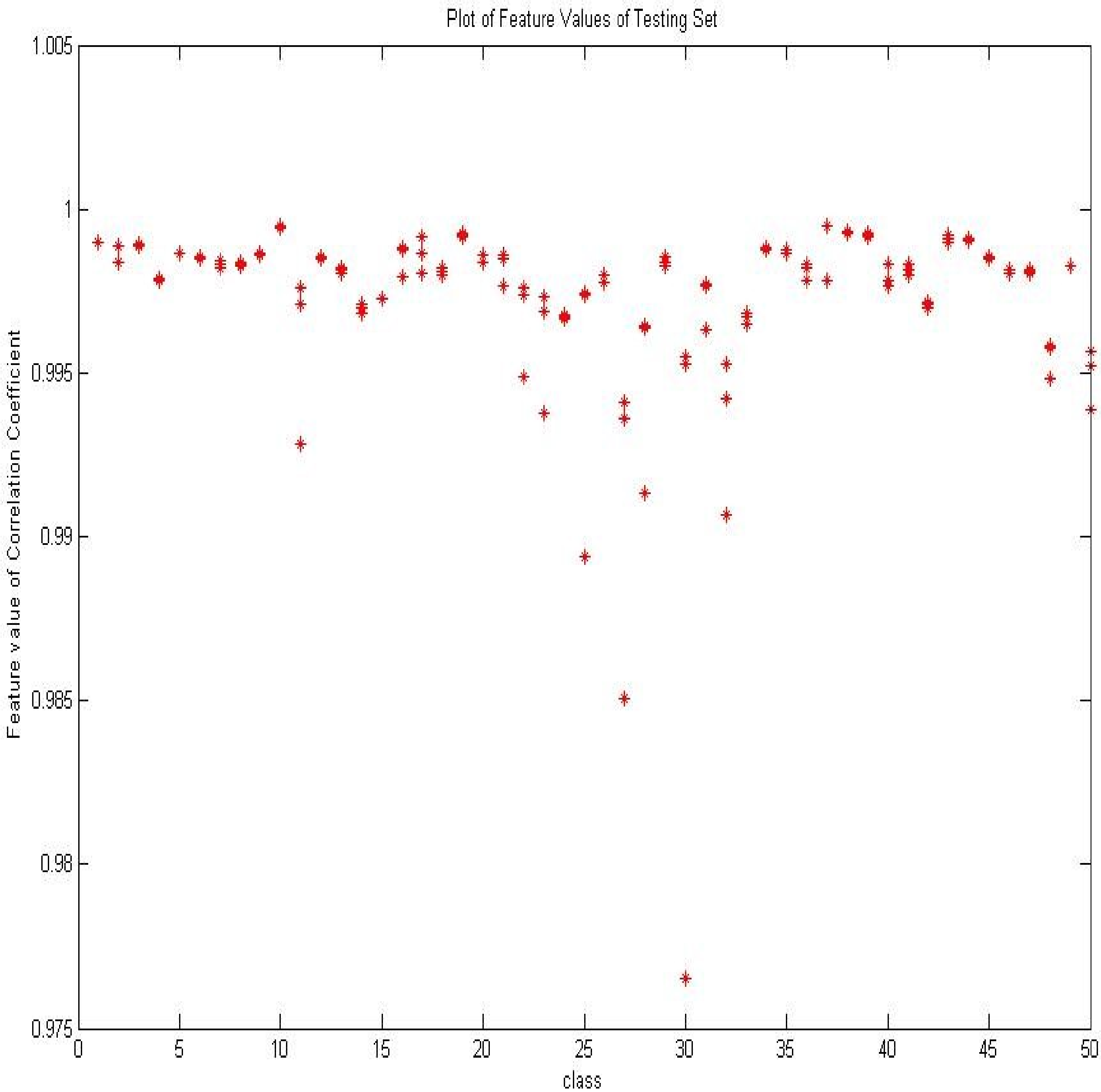


Fig 14 : Plot of R feature Values of Testing Set

4.3.2 *Plot of Feature Value of % of energy of Testing Set*

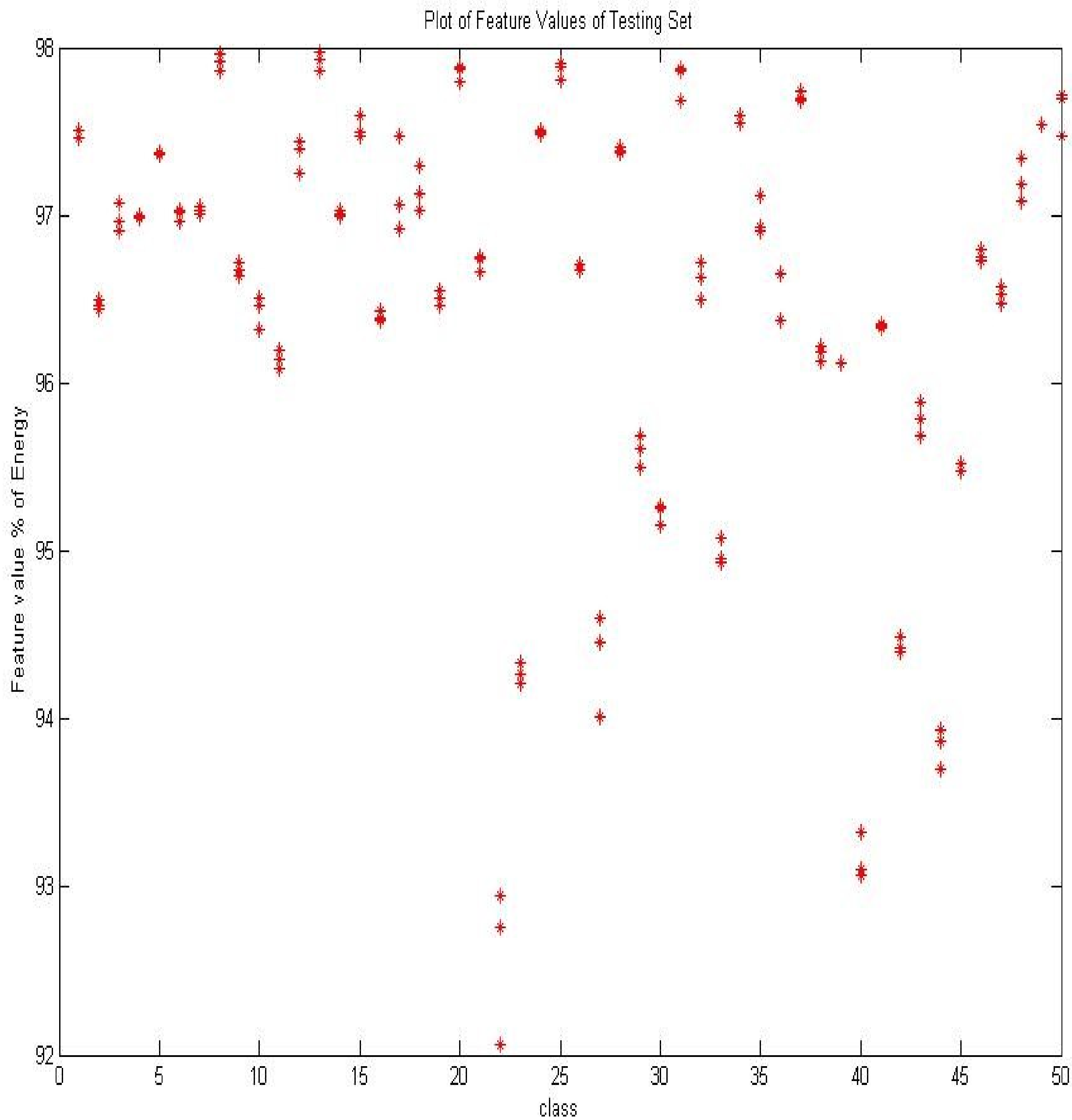


Fig 15 : Plot of Ea feature Values of Testing Set

4.3.3 *Plot of Feature Value of entropy of Testing Set*

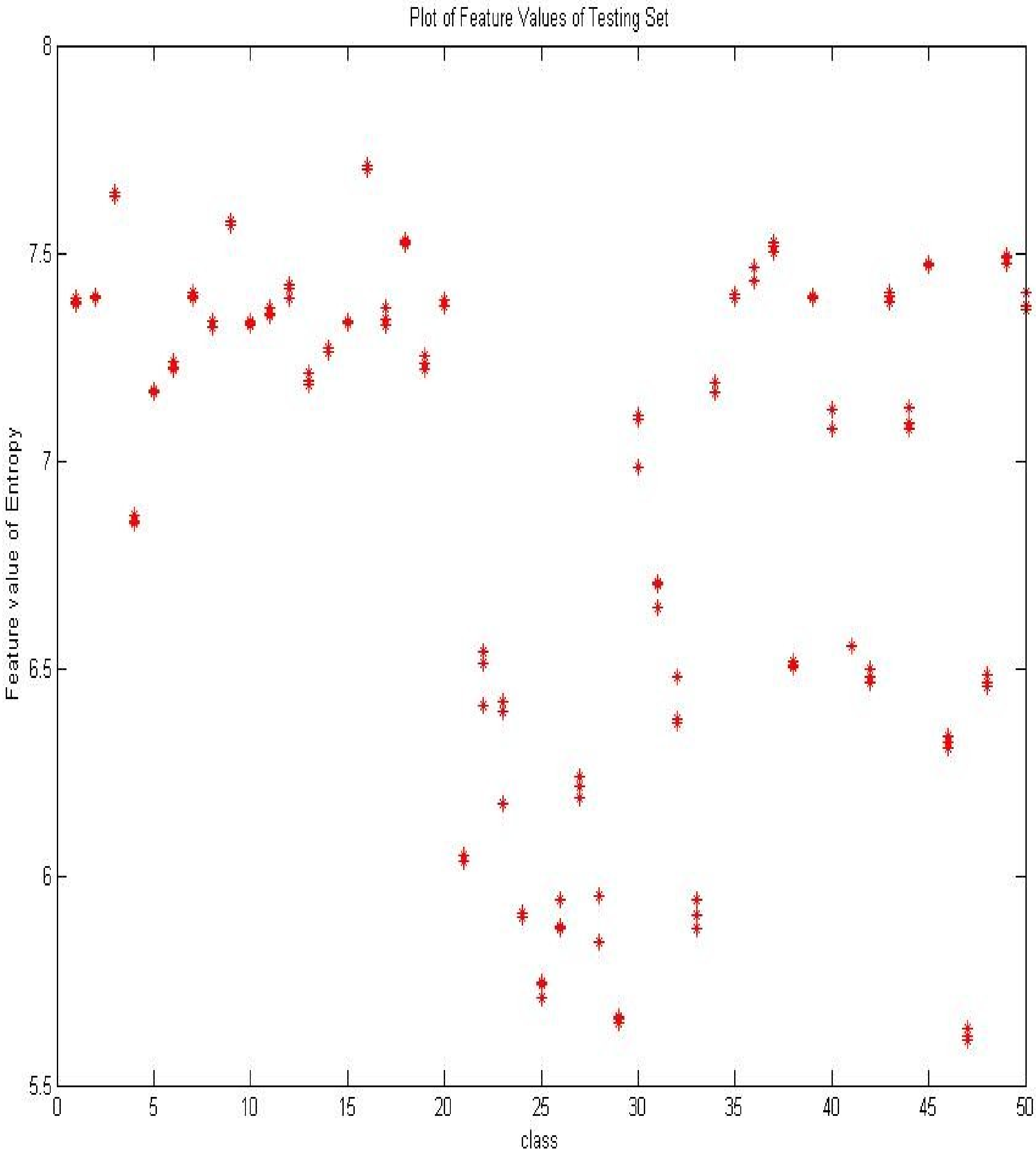


Fig 16 : Plot of e feature Values of Testing Set

4.3.4 *Plot of Feature Value of mean from histogram of Testing Set*

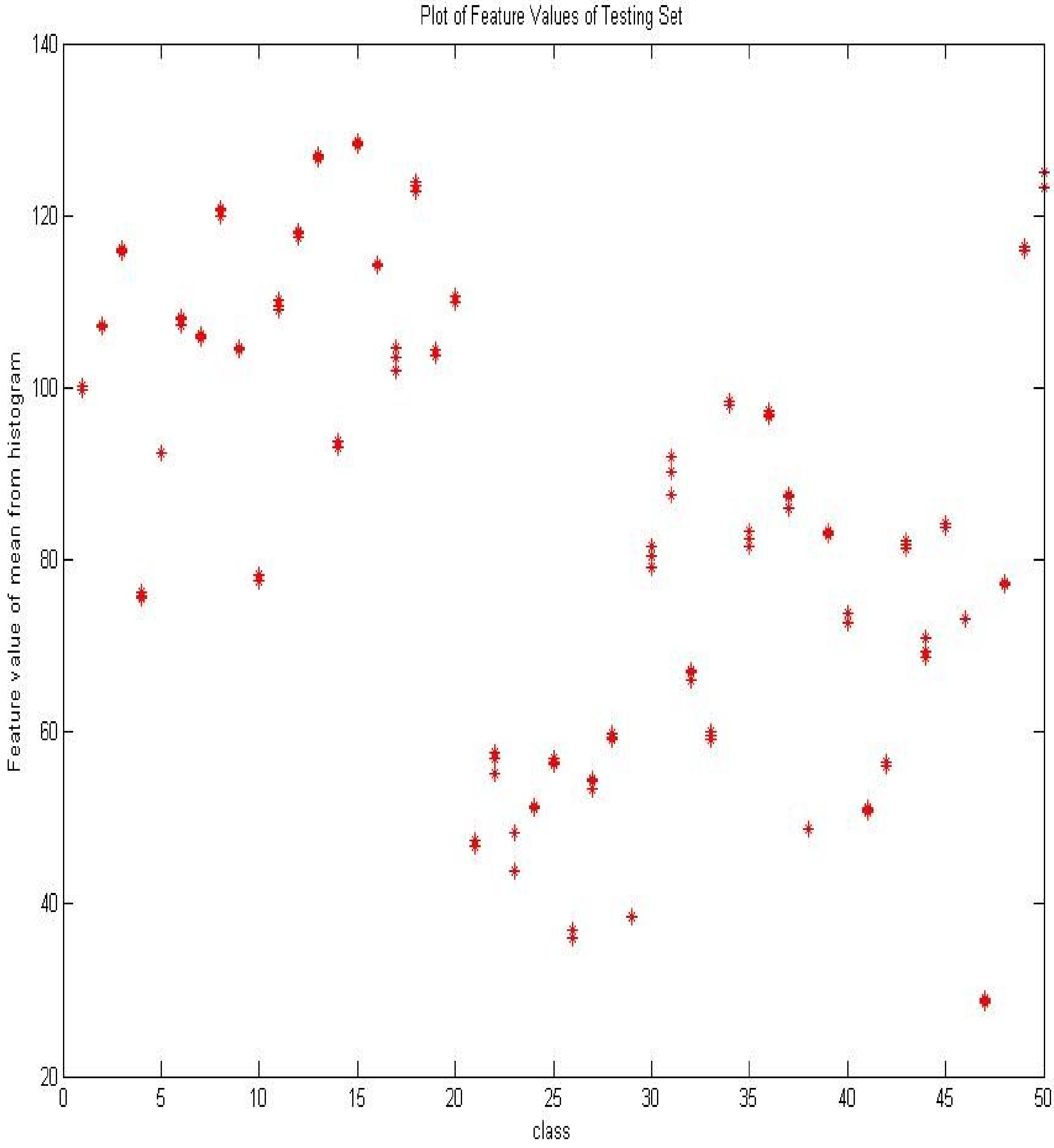


Fig 17 : Plot of mean from histogram feature Values of Testing Set

4.4 *ACCURACY CALCULATION*

Then a D matrix calculating the absolute value of the difference between the feature values of testing face images and the values of matrix M. From D matrix, an ND matrix is calculated the magnitude of vectors. In ND matrix the row indicates the number of classes and the column indicates the number of testing images.

After calculation, the ND Matrix holds the calculation with the class 1 and 50 samples of it, here each column represents each samples of each class and the rows represents the difference between the training sample of each class and testing samples of each classes.

Here the value of ND matrix is given for class 1.

TABLE 3: ND MATRIX FOR CLASS 1

Class NO	Sample test Data1	Sample test Data2	Sample test Data3
1	0.1457	0.2157	0.1257
2	7.3636	7.3686	7.3251
3	16.2251	15.7327	16.1607
4	23.5754	24.1484	24.3454
5	7.5005	7.5730	7.5425
6	8.2719	8.1889	7.5229
7	6.1682	5.9286	6.3311
8	20.9169	20.6078	20.0382
9	4.7215	4.5563	4.5713
10	22.4078	21.7766	22.2742
11	10.3569	9.2776	9.7303
12	18.2847	18.0215	17.7151
13	27.0617	26.9917	26.7383

14	6.8305	6.1513	6.2511
15	28.7186	28.4643	28.3210
16	14.4816	14.3869	14.2884
17	4.7400	2.2118	3.7138
18	23.6781	22.8950	24.0978
19	3.9492	4.5625	4.7102
20	10.7972	10.8308	10.0900
21	52.4830	53.1628	53.1401
22	45.0189	43.1906	42.6295
23	56.1693	51.7340	51.8140
24	48.6453	48.7378	48.4873
25	43.0179	43.3542	43.5766
26	63.0120	63.8514	63.8656
27	45.6136	45.6652	46.6096
28	40.6478	40.4905	40.1303
29	61.5065	61.4074	61.4126
30	19.6577	20.7784	18.4630
31	12.3942	9.7847	7.9387
32	33.9358	32.6736	33.0910
33	40.7692	40.4377	39.9507
34	1.8752	1.5555	1.5844
35	18.2821	17.3887	16.6065
36	2.7662	3.1223	3.3591
37	12.4496	13.9111	12.6082

38	51.2429	51.2713	51.1619
39	16.6197	17.1124	16.8337
40	27.5256	26.5355	26.3133
41	48.9382	49.1497	48.7886
42	43.4407	43.9578	43.5937
43	18.5327	17.7950	18.2245
44	29.2245	30.8324	31.4696
45	15.8953	16.3287	16.1697
46	26.8538	26.7450	26.8755
47	71.3406	71.0353	71.1383
48	22.8329	22.5683	22.5261
49	16.4262	16.4368	16.0855
50	23.3522	25.2419	25.1203

From this ND matrix we can see that the minimum distance is at Class 1 for that sample calculation among all the values. Then we can conclude that the testing sample belongs to Class 1. Then for all testing sample face images the matched samples face images for each class is calculated from cc matrix.

4.5 *RECOGNITION ACCURACY*

Calculating Accuracies...

TABLE 4: RECOGNITION ACCURACY

CLASS NO	CORRECT COUNTS	Recognition Percentages
1	3	100
2	3	100
3	3	100
4	3	100
5	3	100
6	3	100
7	3	100
8	3	100
9	3	100
10	3	100
11	3	100
12	3	100
13	3	100
14	3	100
15	3	100
16	3	100
17	1	33.33
18	3	100
19	1	33.33

20	3	100
21	3	100
22	3	100
23	3	100
24	3	100
25	3	100
26	3	100
27	3	100
28	3	100
29	3	100
30	1	33.33
31	1	33.33
32	3	100
33	3	100
34	3	100
35	2	66.67
36	3	100
37	3	100
38	3	100
39	3	100
40	3	100
41	3	100
42	3	100
43	3	100

44	3	100
45	3	100
46	3	100
47	3	100
48	3	100
49	3	100
50	2	66.67

Overall Recognition Percentage: 93.33%

4.6 *Plot of difference values of testing set*

HERE, NO OF CLASSES =10

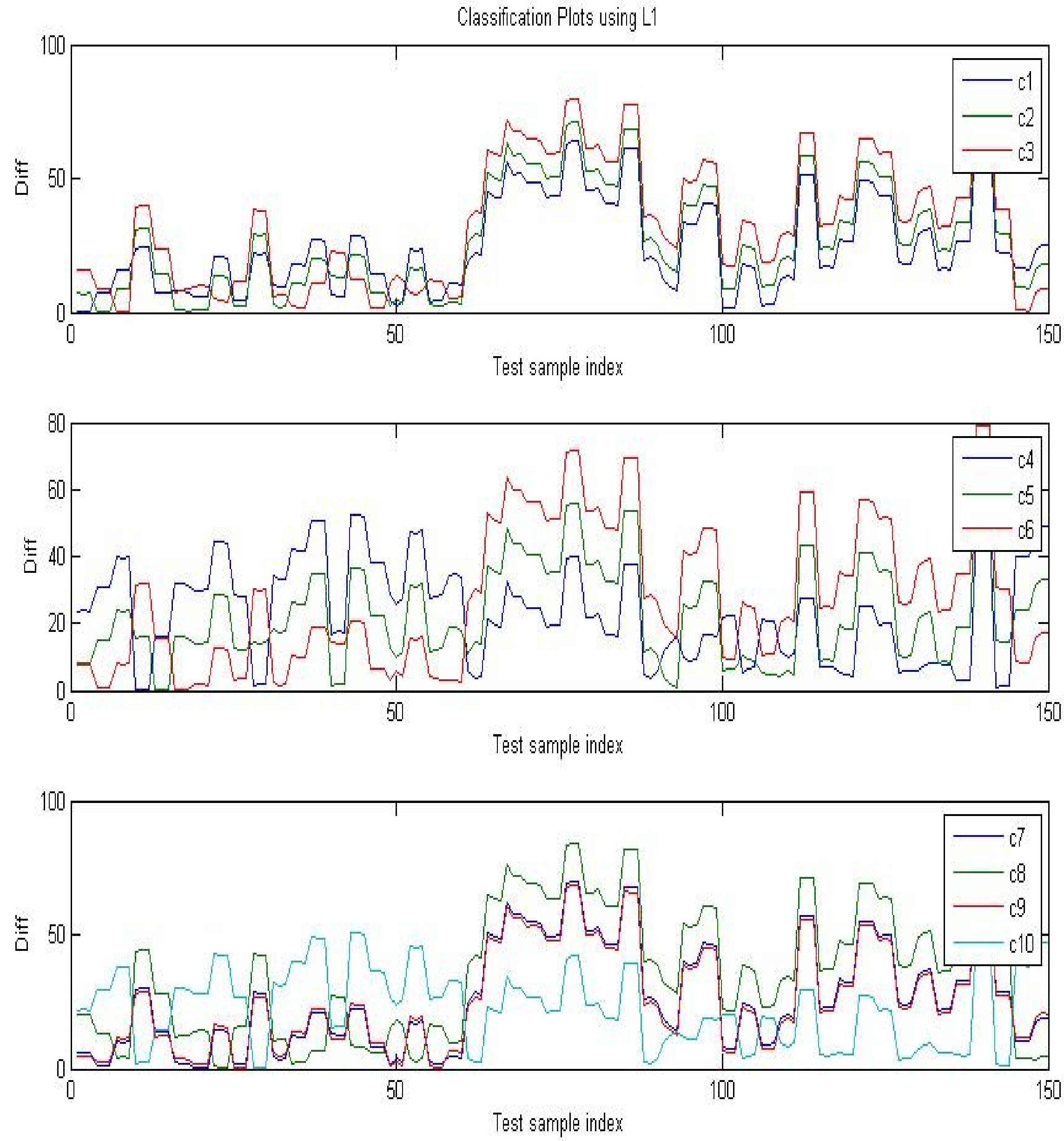


Fig 18 : Plot of Difference Values of Testing Set

CHAPTER 5

ANALYSIS

Using the feature values of wavelet decomposition and mean of histogram for each classes of the Training set and testing set from this study it can obtained 93.33% as overall accuracy. So, after the whole study it can be said that the approach for generating the feature vector is good enough for this dataset.

5.1 *CONTRIBUTIONS*

In the first module, the features of training images from the dataset and the features of the testing images are extracted. This stage uses a set of descriptors to take out the features into vectors. Thus, two groups of feature vectors are created. The second phase of the system has the purpose of comparing the training image features with the set of features of the testing images by Manhattan distance classifier. Finally, the third stage of the system will display the accuracy (recognition rate) result to the user.

5.2 *COMPARISON WITH OTHER WORKS*

I have used the same dataset for other recognition algorithm and got the accuracy of 63.33% for HU's single moment [9] recognition algorithm, 18% for DBC based face recognition [1] algorithm. Compare to the above stated result, study in this approach is given better result.

5.3 *INDIVIDUALITY*

The algorithm which we proposed -
Does not need any background extraction of images. If the train images are not rotated, but the test images are rotated the recognition accuracy is achieving up to 78%.And most importantly, the algorithm does not have the overhead of localizing eyes, nose, mouth etc.

5.4 *ASSUMPTION AND SCOPE*

The system considers fifty classes as the training and testing classes. The implementations are also done likewise. The calculated accuracy level is meant for those fifty classes only. The system cannot tell about the accuracy level of other classes. So the system scope is bounded within those fifty types of classed used in this system.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

Image processing is a vast field with various applications. Recognize the face is a major stream. In this study faces of different kinds of human being have been studied on for their recognition using famous recognition methods like wavelet decomposition and mean from histogram. In the first module, the features of training images from the dataset and the features of the testing images are extracted. This stage uses a set of descriptors to take out the features into vectors. Thus, two groups of feature vectors are created.
The second phase of the system has the purpose of comparing the training image features with the set of features of the testing images by Manhattan distance classifier. Finally, the third stage of the system will display the accuracy (recognition rate) result to the user. Compare to the other existing methods this approach is fast in execution and easy in implementation.

This algorithm is tested on the dataset where the change in train and test data is less. This algorithm can be extended on the dataset where the change in data is vast, not only in expression but also in illumination.

REFERENCES

[1] Jagadeesh H S1, Suresh Babu K2, and Raja K B2 , " DBC based Face Recognition using DWT ", Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.2, Bangalore,India, April 2012

[2] Richa Singh, Mayank Vatsa, and Afzel Noore," Integrated Multilevel Image Fusion and Match Score Fusion of Visible and Infrared Face Images for Robust Face Recognition", Preprint submitted to Elsevier Science, ,USA, 6 July 2007

[3] Soma Biswas, Member, IEEE, Kevin W. Bowyer, Fellow, IEEE, and Patrick J. Flynn, Senior Member, IEEE," Multidimensional Scaling for Matching Low-Resolution Face Images ", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 10,pp(s): 2019 - 2030 ,Notre Dame,OCTOBER 2012,

[4] Manal Abdullah, Majda Wazzan and Sahar Bo-saeed," OPTIMIZING FACE RECOGNITION USING PCA", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, Jeddah,KSA,March 2012

[5] Hafiz Imtiaz and Shaikh Anowarul Fattah," A FACE RECOGNITION SCHEME USING WAVELETBASED DOMINANT FEATURES", Signal & Image Processing : An International Journal (SIPIJ) Vol.2, No.3, Bangladesh, September 2011,

[6] C . N. Ravi Kumar , Bindu . A , "An Efficient Skin Illumination Compensation Model for Efficient Face Detection " , IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on image processing ,Karnataka, PP(s): 3444 – 3449, Publication Year: 2006

[7] Jian Yang , "Color Image Discriminant Models and Algorithms for Face Recognition", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 19, NO. 12, China ,DECEMBER 2008

[8] Jun Luo1, Yong Ma2, Erina Takikawa2, Shihong Lao2, Masato Kawade2, Bao-Liang Lu1 , "PERSON-SPECIFIC SIFT FEATURES FOR FACE RECOGNITION ",Acoustics, Speech and Signal Processing, 2007. ICASSP 2007,Volume: 2, China,pp(s): II-593 - II-596

[9] Rajiv Kapoor, Pallavi Mathur," Face Recognition Using Moments and Wavelets", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 4, India,Jul-Aug 2013, pp. 82-95

[10] Richa Singh, Mayank Vatsa, Arun Ross, Afzel Noore ," Performance Enhancement of 2D Face Recognition via Mosaicing", Appeared in Proc. of 4th IEEE Workshop on Automatic Identification Advanced Technologies (AutoID), (Buffalo, USA), pp. 63-68, October ,2005

[11] Y.K. Eugene and R.G. Johnston, "The Ineffectiveness of the Correlation Coefficient for Image Comparisons", Technical Report LA-UR- 96-2474, Los Alamos, 1996.

[12] Zhao Hong-tu, Yan Jing," The Wavelet Decomposition And Reconstruction Based on The Matlab", Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops(ISECS '10), Guangzhou, P. R. China, 29-31,July 2010, pp. 143-145

Books:

[13] Rafael C. Gonzalez , Richard E. Woods , Digital Image Processing , Publisher: Tom Robbins, ISBN 0-201-18075-8.

[14] Ranjan Parekh , "Principles of Multimedia" , Publisher: Tata McGrow-Hill , ISBN-10: 0-07-058833-3., 2006

[15] Rhys Lewis, Practical Image Processing, Ellis Horwood Limited, 1990. ISBN 0-13-6383525-0. p 90-91.

[16] William Pratt, Digital Image Processing, Second Edition, John Wiley & Sons, Inc., 1991. ISBN 0-471-85766-1. p. 561-563.

[17] Oge Marques, Practical Image and Video Processing Using MATLAB, Publisher: Wiley-IEEE Press, Second Edition, Volume 1, Academic Press, 1982. ISBN 978-0-470-04815-3, September 21, 2011

WEBSITES:

[18] http://cswww.essex.ac.uk/mv/allfaces/faces94.html
Collection of face images

[19] http://www.mathworks.com/

[20]http://nptel.iitk.ac.in/courses/Webcourse-contents/IIT-KANPUR/Digi_Img_Pro/chapter_8/8_16.html

[21] http://www.johnloomis.org/ece563/notes/basics/entropy/entropy.html
[22] http://en.wikipedia.org/wiki/Feature_(computer_vision)
[23] http://www.imageprocessingbasics.com/rgb-to-grayscale/

APPENDIX A

CODE SNIPPETS

A1.START

```
clear all;
clc;
path = 'data/';
ext = '.jpg';
train;
test;
accu;
```

A2.TRAIN

```
NC = 50;  %No. of classes
NT = 4;  %No. of items in training set
NS = 3;  %No. of items in testing set
fprintf('\nGenerating Training Vectors ...\n\n')
for CL = 1 : NC
   for SA = 1 : NT
      class = num2str(CL);
      sample = num2str(SA);
      I=imread(strcat(path,class,' (',sample,')',ext));
      process_image;
      T(SA,:,CL)=E;
       end
   fprintf('Class %d Completed\n', CL)
end
```

A3.TEST

```
fprintf('\nGenerating Testing Vectors ...\n\n')
for CL = 1 : NC
   for SA = NT+1 : NT+NS
      class = num2str(CL);
      sample = num2str(SA);
      I=imread(strcat(path,class,' (',sample,')',ext));
```

```
        process_image;
        S(SA-NT,:,CL )=E;
        end
    fprintf('Class %d Completed\n', CL)
end
```

A4.ACCURACY

```
fprintf('\n\nCalculating Accuracies ...\n')

NC=size(T,3);   %no. of classes
LV=size(T,2);   %length of feature vector
NT=size(T,1);   %no. of train files/class
NS=size(S,1);   %no. of test files/class

M=zeros(NC,LV);
for i=1:NC
   M(i,:)=mean(T(:,1:LV,i));
end

D=zeros(NS,LV,NC*NC);

for i=1:NS
   for j=1:NC
      for k=1:NC
         D(i,:,k+(j-1)*NC)=abs(S(i,:,j)-M(k,:));
      end
   end
end

%calc magnitude of vectors
ND=zeros(NC,NS,NC);
for i=1:NS
   for j=1:NC
      for k=1:NC
         ND(j,i,k)=norm(D(i,:,k+(j-1)*NC));
      end
   end
end
```

```
cc=zeros(1,NC); %correct count

for i=1:NC
   for j=1:NS
         if min(ND(i,j,:)) == ND(i,j,i)
            cc(i)=cc(i)+1;
         end
   end
end
pac=(cc/NS)*100;
oac = mean(pac);
fprintf('\nCorrect Counts : ')
fprintf('%d ', cc)
fprintf('\nClass Recognition Percentages : ')
fprintf('%.2f, ', pac)
fprintf('\nOverall Recognition Percentage : ')
fprintf('%.2f ', oac)
fprintf('\n')
```

A5.PROCESS_IMAGE

```
I=rgb2gray(I);
[pixelCount grayLevels] = imhist(I);
meanGL = sum(pixelCount .* grayLevels) / sum(pixelCount);
e=entropy(I);
I=double(I);
J=medfilt2(I);
R=corr2(I,J);
[C,L]=wavedec(I,4,'sym4');
[Ea,Ed]=wenergy(C,L);
E=[R Ea e meanGL];
```

A6.PLOT

```
f=[]; f=1:NC*NS;
DC1=[ND(1,:,1),ND(2,:,1),   ND(3,:,1),   ND(4,:,1),   ND(5,:,1),   ND(6,:,1),   ND(7,:,1),
ND(8,:,1), ND(9,:,1), ND(10,:,1), ...
    ND(11,:,1),ND(12,:,1),  ND(13,:,1),  ND(14,:,1),  ND(15,:,1),  ND(16,:,1),  ND(17,:,1),
ND(18,:,1), ND(19,:,1), ND(20,:,1), ...
```

ND(21,:,1),ND(22,:,1), ND(23,:,1), ND(24,:,1), ND(25,:,1), ND(26,:,1), ND(27,:,1), ND(28,:,1), ND(29,:,1), ND(30,:,1), ...
ND(31,:,1),ND(32,:,1), ND(33,:,1), ND(34,:,1), ND(35,:,1), ND(36,:,1), ND(37,:,1), ND(38,:,1), ND(39,:,1), ND(40,:,1), ...
ND(41,:,1),ND(42,:,1), ND(43,:,1), ND(44,:,1), ND(45,:,1), ND(46,:,1), ND(47,:,1), ND(48,:,1), ND(49,:,1), ND(50,:,1), ...
];
DC2=[ND(1,:,2),ND(2,:,2), ND(3,:,2), ND(4,:,2), ND(5,:,2), ND(6,:,2), ND(7,:,2), ND(8,:,2), ND(9,:,2), ND(10,:,2), ...
ND(11,:,2),ND(12,:,2), ND(13,:,2), ND(14,:,2), ND(15,:,2), ND(16,:,2), ND(17,:,2), ND(18,:,2), ND(19,:,2), ND(20,:,2), ...
ND(21,:,2),ND(22,:,2), ND(23,:,2), ND(24,:,2), ND(25,:,2), ND(26,:,2), ND(27,:,2), ND(28,:,2), ND(29,:,2), ND(30,:,2), ...
ND(31,:,2),ND(32,:,2), ND(33,:,2), ND(34,:,2), ND(35,:,2), ND(36,:,2), ND(37,:,2), ND(38,:,2), ND(39,:,2), ND(40,:,2), ...
ND(41,:,2),ND(42,:,2), ND(43,:,2), ND(44,:,2), ND(45,:,2), ND(46,:,2), ND(47,:,2), ND(48,:,2), ND(49,:,2), ND(50,:,2), ...
];
DC3=[ND(1,:,3),ND(2,:,3), ND(3,:,3), ND(4,:,3), ND(5,:,3), ND(6,:,3), ND(7,:,3), ND(8,:,3), ND(9,:,3), ND(10,:,3), ...
ND(11,:,3),ND(12,:,3), ND(13,:,3), ND(14,:,3), ND(15,:,3), ND(16,:,3), ND(17,:,3), ND(18,:,3), ND(19,:,3), ND(20,:,3), ...
ND(21,:,3),ND(22,:,3), ND(23,:,3), ND(24,:,3), ND(25,:,3), ND(26,:,3), ND(27,:,3), ND(28,:,3), ND(29,:,3), ND(30,:,3), ...
ND(31,:,3),ND(32,:,3), ND(33,:,3), ND(34,:,3), ND(35,:,3), ND(36,:,3), ND(37,:,3), ND(38,:,3), ND(39,:,3), ND(40,:,3), ...
ND(41,:,3),ND(42,:,3), ND(43,:,3), ND(44,:,3), ND(45,:,3), ND(46,:,3), ND(47,:,3), ND(48,:,3), ND(49,:,3), ND(50,:,3), ...
];
DC4=[ND(1,:,4),ND(2,:,4), ND(3,:,4), ND(4,:,4), ND(5,:,4), ND(6,:,4), ND(7,:,4), ND(8,:,4), ND(9,:,4), ND(10,:,4), ...
ND(11,:,4),ND(12,:,4), ND(13,:,4), ND(14,:,4), ND(15,:,4), ND(16,:,4), ND(17,:,4), ND(18,:,4), ND(19,:,4), ND(20,:,4), ...
ND(21,:,4),ND(22,:,4), ND(23,:,4), ND(24,:,4), ND(25,:,4), ND(26,:,4), ND(27,:,4), ND(28,:,4), ND(29,:,4), ND(30,:,4), ...
ND(31,:,4),ND(32,:,4), ND(33,:,4), ND(34,:,4), ND(35,:,4), ND(36,:,4), ND(37,:,4), ND(38,:,4), ND(39,:,4), ND(40,:,4), ...
ND(41,:,4),ND(42,:,4), ND(43,:,4), ND(44,:,4), ND(45,:,4), ND(46,:,4), ND(47,:,4), ND(48,:,4), ND(49,:,4), ND(50,:,4), ...
];

```
DC5=[ND(1,:,5),ND(2,:,5), ND(3,:,5), ND(4,:,5), ND(5,:,5), ND(6,:,5), ND(7,:,5), ND(8,:,5), ND(9,:,5), ND(10,:,5), ...
    ND(11,:,5),ND(12,:,5), ND(13,:,5), ND(14,:,5), ND(15,:,5), ND(16,:,5), ND(17,:,5), ND(18,:,5), ND(19,:,5), ND(20,:,5), ...
    ND(21,:,5),ND(22,:,5), ND(23,:,5), ND(24,:,5), ND(25,:,5), ND(26,:,5), ND(27,:,5), ND(28,:,5), ND(29,:,5), ND(30,:,5), ...
    ND(31,:,5),ND(32,:,5), ND(33,:,5), ND(34,:,5), ND(35,:,5), ND(36,:,5), ND(37,:,5), ND(38,:,5), ND(39,:,5), ND(40,:,5), ...
    ND(41,:,5),ND(42,:,5), ND(43,:,5), ND(44,:,5), ND(45,:,5), ND(46,:,5), ND(47,:,5), ND(48,:,5), ND(49,:,5), ND(50,:,5), ...
    ];
DC6=[ND(1,:,6),ND(2,:,6), ND(3,:,6), ND(4,:,6), ND(5,:,6), ND(6,:,6), ND(7,:,6), ND(8,:,6), ND(9,:,6), ND(10,:,6), ...
    ND(11,:,6),ND(12,:,6), ND(13,:,6), ND(14,:,6), ND(15,:,6), ND(16,:,6), ND(17,:,6), ND(18,:,6), ND(19,:,6), ND(20,:,6), ...
    ND(21,:,6),ND(22,:,6), ND(23,:,6), ND(24,:,6), ND(25,:,6), ND(26,:,6), ND(27,:,6), ND(28,:,6), ND(29,:,6), ND(30,:,6), ...
    ND(31,:,6),ND(32,:,6), ND(33,:,6), ND(34,:,6), ND(35,:,6), ND(36,:,6), ND(37,:,6), ND(38,:,6), ND(39,:,6), ND(40,:,6), ...
    ND(41,:,6),ND(42,:,6), ND(43,:,6), ND(44,:,6), ND(45,:,6), ND(46,:,6), ND(47,:,6), ND(48,:,6), ND(49,:,6), ND(50,:,6), ...
    ];
DC7=[ND(1,:,7),ND(2,:,7), ND(3,:,7), ND(4,:,7), ND(5,:,7), ND(6,:,7), ND(7,:,7), ND(8,:,7), ND(9,:,7), ND(10,:,7), ...
    ND(11,:,7),ND(12,:,7), ND(13,:,7), ND(14,:,7), ND(15,:,7), ND(16,:,7), ND(17,:,7), ND(18,:,7), ND(19,:,7), ND(20,:,7), ...
    ND(21,:,7),ND(22,:,7), ND(23,:,7), ND(24,:,7), ND(25,:,7), ND(26,:,7), ND(27,:,7), ND(28,:,7), ND(29,:,7), ND(30,:,7), ...
    ND(31,:,7),ND(32,:,7), ND(33,:,7), ND(34,:,7), ND(35,:,7), ND(36,:,7), ND(37,:,7), ND(38,:,7), ND(39,:,7), ND(40,:,7), ...
    ND(41,:,7),ND(42,:,7), ND(43,:,7), ND(44,:,7), ND(45,:,7), ND(46,:,7), ND(47,:,7), ND(48,:,7), ND(49,:,7), ND(50,:,7), ...
    ];
DC8=[ND(1,:,8),ND(2,:,8), ND(3,:,8), ND(4,:,8), ND(5,:,8), ND(6,:,8), ND(7,:,8), ND(8,:,8), ND(9,:,8), ND(10,:,8), ...
    ND(11,:,8),ND(12,:,8), ND(13,:,8), ND(14,:,8), ND(15,:,8), ND(16,:,8), ND(17,:,8), ND(18,:,8), ND(19,:,8), ND(20,:,8), ...
    ND(21,:,8),ND(22,:,8), ND(23,:,8), ND(24,:,8), ND(25,:,8), ND(26,:,8), ND(27,:,8), ND(28,:,8), ND(29,:,8), ND(30,:,8), ...
```

```
    ND(31,:,8),ND(32,:,8), ND(33,:,8), ND(34,:,8), ND(35,:,8), ND(36,:,8), ND(37,:,8),
ND(38,:,8), ND(39,:,8), ND(40,:,8), ...
    ND(41,:,8),ND(42,:,8), ND(43,:,8), ND(44,:,8), ND(45,:,8), ND(46,:,8), ND(47,:,8),
ND(48,:,8), ND(49,:,8), ND(50,:,8), ...
    ];
DC9=[ND(1,:,9),ND(2,:,9), ND(3,:,9), ND(4,:,9), ND(5,:,9), ND(6,:,9), ND(7,:,9),
ND(8,:,9), ND(9,:,9), ND(10,:,9), ...
    ND(11,:,9),ND(12,:,9), ND(13,:,9), ND(14,:,9), ND(15,:,9), ND(16,:,9), ND(17,:,9),
ND(18,:,9), ND(19,:,9), ND(20,:,9), ...
    ND(21,:,9),ND(22,:,9), ND(23,:,9), ND(24,:,9), ND(25,:,9), ND(26,:,9), ND(27,:,9),
ND(28,:,9), ND(29,:,9), ND(30,:,9), ...
    ND(31,:,9),ND(32,:,9), ND(33,:,9), ND(34,:,9), ND(35,:,9), ND(36,:,9), ND(37,:,9),
ND(38,:,9), ND(39,:,9), ND(40,:,9), ...
    ND(41,:,9),ND(42,:,9), ND(43,:,9), ND(44,:,9), ND(45,:,9), ND(46,:,9), ND(47,:,9),
ND(48,:,9), ND(49,:,9), ND(50,:,9), ...
    ];
DC10=[ND(1,:,10),ND(2,:,10), ND(3,:,10), ND(4,:,10), ND(5,:,10), ND(6,:,10),
ND(7,:,10), ND(8,:,10), ND(9,:,10), ND(10,:,10), ...
    ND(11,:,10),ND(12,:,10), ND(13,:,10), ND(14,:,10), ND(15,:,10), ND(16,:,10),
ND(17,:,10), ND(18,:,10), ND(19,:,10), ND(20,:,10), ...
    ND(21,:,10),ND(22,:,10), ND(23,:,10), ND(24,:,10), ND(25,:,10), ND(26,:,10),
ND(27,:,10), ND(28,:,10), ND(29,:,10), ND(30,:,10), ...
    ND(31,:,10),ND(32,:,10), ND(33,:,10), ND(34,:,10), ND(35,:,10), ND(36,:,10),
ND(37,:,10), ND(38,:,10), ND(39,:,10), ND(40,:,10), ...
    ND(41,:,10),ND(42,:,10), ND(43,:,10), ND(44,:,10), ND(45,:,10), ND(46,:,10),
ND(47,:,10), ND(48,:,10), ND(49,:,10), ND(50,:,10), ...
    ];

figure
subplot(311), plot(f, DC1, f, DC2, f, DC3); xlabel('Test sample index'); ylabel('Diff');
legend('c1', 'c2', 'c3');
title('Classification Plots using L1')
subplot(312), plot(f, DC4, f, DC5, f, DC6); xlabel('Test sample index'); ylabel('Diff');
legend('c4', 'c5', 'c6');
subplot(313), plot(f, DC7, f, DC8, f, DC9, f, DC10); xlabel('Test sample index');
ylabel('Diff'); legend('c7', 'c8', 'c9', 'c10');
```