A

MAJOR PROJECT REPORT ON

LEAF DISEASE IDENTIFICATION

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

K. ADITHI **:18Q91A0561**

Under the guidance of

Dr. T. SUNIL
Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MALLA REDDY COLLEGE OF ENGINEERING
(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)
Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution
Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100

2021 - 2022

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Major Project report on "LEAF DISEASE IDENTIFICATION" is successfully done by the following students of Department of Computer Science & Engineering of our college in partial fulfilment of the requirement for the award of B.Tech degree in the year 2021-2022. The results embodied in this report have not been submitted to any other University for the award of any diploma or degree.

K. ADITHI **:18Q91A0561**

INTERNAL GUIDE	HOD	PRINCIPAL
Dr. T. Sunil	Dr. G. Radha Devi	Dr. M. Sreedhar Reddy
Professor	Assoc. Professor	Professor

Submitted for the viva voice examination held on: _____

Internal Examiner **External Examiner**

DECLARATION

I, K.Adithi, with Regd.no 18Q91A0561, are hereby declaring that the major project report entitled **"Leaf Disease Identification"** has done by us under the guidance of **Dr. T. Sunil** Professor, Department of CSE is submitted in the partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING.**

The Results embeded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Signature of the Candidate

K.Adithi **18Q91A0561**

DATE:

PLACE: Maisammaguda

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank **Sri Ch. Malla Reddy**, our Founder, Chairman **MRGI**, **Sri Ch. Mahender Reddy**, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of **Dr. M. Sreedhar Reddy** our beloved principal for his kind co-operation and encouragement in bringing out this task.

We would like to thank **Dr. T. V. Reddy** our vice principal, **Dr. G. Radha Devi** HOD, CSE Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank **Dr. T. Sunil** Professor our internal guide, for her valuable suggestions and guidance during the exhibition and completion of this project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

K.Adithi **18Q91A0561**

ABSTRACT

Agricultural productivity is that thing on which Indian Economy highly depends. This is the one of the reasons that disease detection in plants plays an important role in agriculture field, as having disease in plants are quite natural. If proper care is not taken in this area then it causes serious effects on plants and due to which respective product quality, quantity or productivity is affected. Detection of plant disease through some automatic technique is beneficial as it reduces a large work of monitoring in big farms of crops, and at very early stage itself it detects the symptoms of diseases means when they appear on plant leaves. This paper presents an algorithm for image segmentation technique used for automatic detection as well as classification of plant leaf diseases and survey on different diseases classification techniques that can be used for plant leaf disease detection. Image segmentation, which is an important aspect for disease detection in plant leaf disease, is done by using Neural Network.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREEN SHOTS

CHAPTER-1
INTRODUCTION

1.1 Introduction:

Indian economy is dependent of agricultural productivity. Over 70% of rural homes depend on agriculture. Agriculture pays about 17% to the total GDP and provides employment to over 60% of the population. Therefore detection of plant diseases plays a vital key role in the arena of agriculture. Indian agriculture is composed of many crops like rice, wheat. Indian farmers also grow sugarcane, oilseeds, potatoes and non-food items like coffee, tea, cotton, rubber. All these crops grow based on strength of leaves and roots. There are things that lead to different disease for the plant leaves, which spoiled crops and finally it will effect on economy of the country

1.2 Overview:

The image processing technique had been utilized to automatically detect and diagnosis plant disease. In this work, threshold technique was used to remove unwanted pixels and classify the type of disease. The main purpose is to identify the disease affected in the leaf.

1.3 Scope Of Project:

The main contributions of this project therefore are

- Data Analysis
- Dataset Pre-processing
- Training the Model
- Testing of Dataset

1.4 Domain Overview:

Deep neural networks are now the state-of-the-art machine learning models across a variety of areas, from image analysis to natural language processing, and widely deployed in academia and industry. These developments have a huge potential for medical imaging technology, medical data analysis, medical diagnostics and healthcare in general, slowly being realized. We provide a short overview of recent

advances and some associated challenges in machine learning applied to medical image processing and image analysis.

Long before deep learning was used, traditional machine learning methods were mainly used. Such as Decision Trees, SVM, Naïve Bayes Classifier and Logistic Regression.

These algorithms are also called flat algorithms. Flat here means that these algorithms can not normally be applied directly to the raw data (such as .csv, images, text, etc.). We need a preprocessing step called Feature Extraction.

The result of Feature Extraction is a representation of the given raw data that can now be used by these classic machine learning algorithms to perform a task. For example, the classification of the data into several categories or classes.

Feature Extraction is usually quite complex and requires detailed knowledge of the problem domain. This pre-processing layer must be adapted, tested and refined over several iterations for optimal results.

On the other side are the artificial neural networks of Deep Learning. These do not need the Feature Extraction step. The layers are able to learn an implicit representation of the raw data directly and on their own. Here, a more and more abstract and compressed representation of the raw data is produced over several layers of artificial neural-nets. This compressed representation of the input data is then used to produce the result. The result can be, for example, the classification of the input data into different classes.

CHAPTER-2
LITERATURE SURVEY

2.1 Literature Survey:

[1]Sachin D. Khirade, A.B Patil, "Plant Disease Detection Using Image Processing", International Conference on Computing Communication Control and Automation", 2015.

Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time. Hence, image processing is used for the detection of plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification. This paper discussed the methods used for the detection of plant diseases using their leaves images. This paper discussed various techniques to segment the disease part of the plant. This paper also discussed some Feature extraction and classification techniques to extract the features of infected leaf and the classification of plant diseases. The accurately detection and classification of the plant disease is very important for the successful cultivation of crop and this can be done using image processing. This paper discussed various techniques to segment the disease part of the plant. This paper also discussed some Feature extraction and classification techniques to extract the features of infected leaf and the classification of plant diseases. The use of ANN methods for classification of disease in plants such as self- organizing feature map, back propagation algorithm, SVMs etc. can be efficiently used. From these methods, we can accurately identify and classify various plant diseases using image processing technique.

[2]Prof. Sanjay B. Dhaygude, Mr.Nitin P.Kumbhar, "Agricultural plant Leaf Disease Detection Using Image Processing", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 1, January 2013.

The application of texture statistics for detecting the plant leaf disease has been explained Firstly by color transformation structure RGB is converted into HSV space because HSV is a good color descriptor. Masking and removing of green pixels with pre-computed threshold level. Then in the next step segmentation is performed using 32X32 patch size and obtained useful segments. These segments are used for texture analysis by color co-occurrence matrix. Finally if texture parameters are compared to texture parameters of normal leaf.

[3]Amandeep Singh ,Maninder Lal Singh, "Automated Color Prediction of Paddy Crop Leaf using Image Processing", International Conference on Technological Innovations in ICT for Agriculture and Rural Development (TIAR 2015), 2015.

The most significant challenge faced during the work was capturing the quality images with maximum detail of the leaf color. It is very typical task to get the image with all the details within a procesable memory. Such images are formed a through high resolution and thus are of 6-10MB of size. This was handled by using a Nikon made D5200 camera which served the task very well. Second challenge faced was to get rid of illumination conditions as from the start to the end of paddy crop season, illumination varies a lot even when the image acquiring time is fixed. However the solution to this is variable user defined thresholding and making necessary adjustments to the shades of LCC.

[4]M.Malathi, K.Aruli , S.Mohamed Nizar, A.Sagaya Selvaraj, "A Survey on Plant Leaf Disease Detection Using Image Processing Techniques",International Research Journal of Engineering and Technology (IRJET),Volume: 02 Issue: 09, Dec 2015.

They provides survey on plant leaf disease detection using image processing techniques. Disease in crops causes significant reduction in quantity and quality of the agricultural product. Identification of symptoms of disease by naked eye is difficult for farmer. Crop protection especially in large farms is done by using computerized image processing technique that can detect diseased leaf using color information of leaves.Depending on the applications, many image processing technique has been introduced to solve the problems by pattern recognition and some automatic classification tools. In the next section this papers present a survey of those proposed systems in meaningful way.There are many methods in automated or computer vision for disease detection and classification but still there is lack in this research topic. All the disease cannot be identified using single method.

[5]MalvikaRanjan, Manasi Rajiv Weginwar, NehaJoshi, Prof.A.B. Ingole, "detection and classification of leaf disease using artificial neural network", International Journal of Technical Research and Applications, 2015.

Describes a diagnosis process that is mostly visual and requires precise judgment and also scientific methods. Image of diseased leaf is captured .As the result of segmentation Color HSV features are extracted. Artificial neural network (ANN) is then trained to distinguish the healthy and diseased samples. ANN classification performance is 80% better in accuracy.

2.2 Basics of Image Processing:

Fundamentals Of Digital Image

An image is a two-dimensional picture, which has a similar appearance to some subject, usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes,microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.

Fig 2.2.1: Color image to Grayscale Conversion Process

An image is a rectangular grid of pixels. It has a definite height and a definite width blueness, and the remaining eight bits the transparency of the pixel. counted in pixels. Each pixel is square and has a fixedsize on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

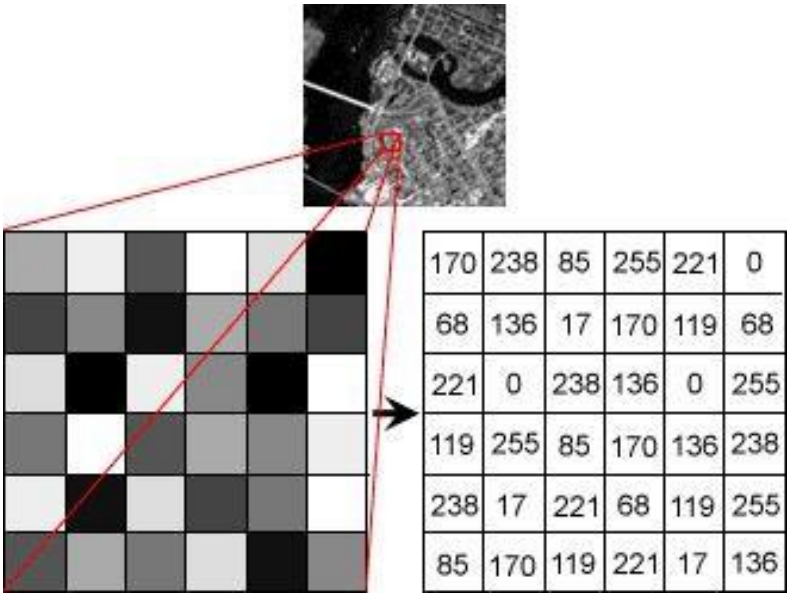


Fig 2.2.2 : Gray Scale Image Pixel Value Analysis

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel,the next eight bits the greenness, the next eight bits the

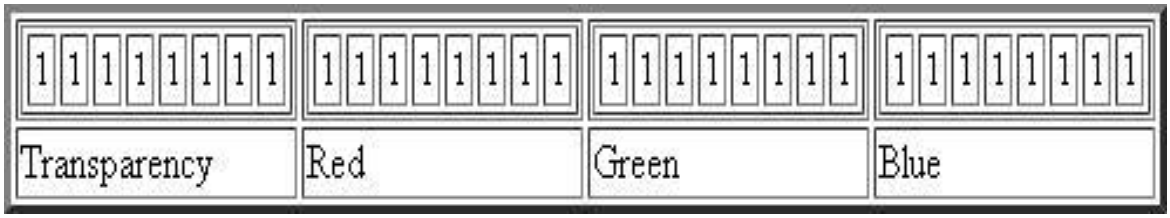


Fig 2.2.3 : BIT Transferred for Red, Green and Blue plane (24bit=8 bit red;8-bit)

2.2.1 Image File Sizes:

Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

2.2.2 Image File Formats:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images.Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images the Internet.In addition to straight image formats.

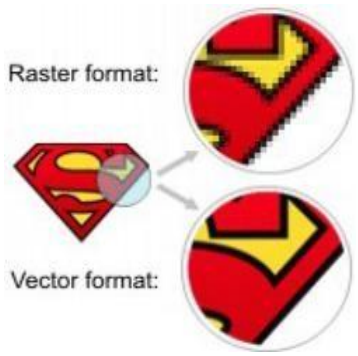


Fig 2.2.4 : Horizontal and Vertical Process

2.2.3 Image Processing:

Digital image processing, the manipulation of images by computer, is a relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of image with varying degrees of success. The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing, like other glamor fields, suffers from myths, missed-connections, mis- understandings and mis-information. It is a vast umbrella under which diverse aspects of optics, electronics, mathematics, photography, graphics and computer technology. It is truly a multidisciplinary endeavor plowed with imprecise jargon.

Several factors combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

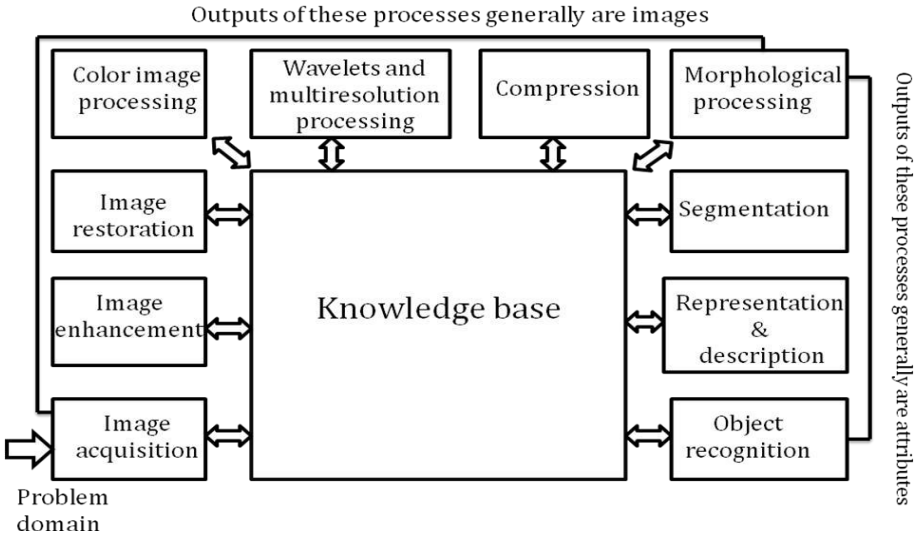


Fig 2.2.5 : Basics steps of image Processing

2.2.4 Image Acquisition:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be a monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. The image sensor could also be a line scan camera that produces a single image line at a time. In this case, the object's motion past the line.

Fig 2.2.6 : Digital camera

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it producesare determined by the application.

2.2.5 Image Enhancement:

Image enhancement: is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of an interesting image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 2.2.7 : Image enhancement process for Gray Scale Image and Color Image using Histogram

2.2.6 Image Restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Fig 2.2.8 : Noise image **Image Enhancement**

2.2.7 Color Image Processing:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. Thissecond factor is particularly important in manual image analys

Fig 2.2.9 : gray Scale image **Color Image**

2.2.8 Segmentation:

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

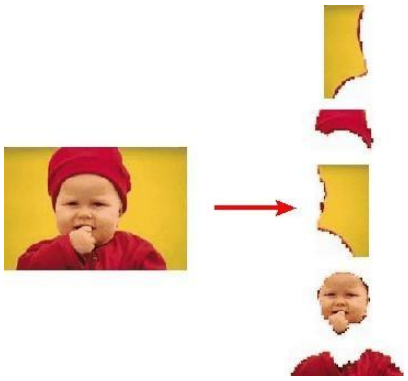


Fig 2.2.10 : Image Segment Process

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed. Digital image is defined as a two dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called intensity or gray level of the image at that point. The field of digital image processing refers to processing digital images by means ofa digital computer. The digital image is composed of a finite number of elements, each of which has a particular location and value. The elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used.

2.2.9 Image Compression Type:

Digital Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data. From the mathematical viewpoint, this amounts to transforming a 2D pixel array into a statically uncorrelated data set. The data redundancy is not an abstract concept but a mathematically quantifiable entity. If n1 and n2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D set (the one characterized by n1) can be defined as,of the first data Where C_R called compression ratio [2]. It is defined as $C_R = n2$In image compression, three basic data redundancies can be identified and exploited: Coding redundancy, interpixel redundancy, and phychovisal redundancy.Image compression is achieved when one or more of these redundancies are reduced or eliminated. The image compression is mainly used for image transmission and storage. Image transmission applications are in broadcast television; remote sensing via satellite, air-craft, radar, or sonar; teleconferencing; computer communications; and facsimile transmission. Image storage is required most commonly for educational and business documents, medical images that arise in computer tomography (CT), magnetic resonance imaging (MRI) and digital radiology, motion pictures, satellite images, weather maps, geological surveys, and so on.

Image Compression Model:

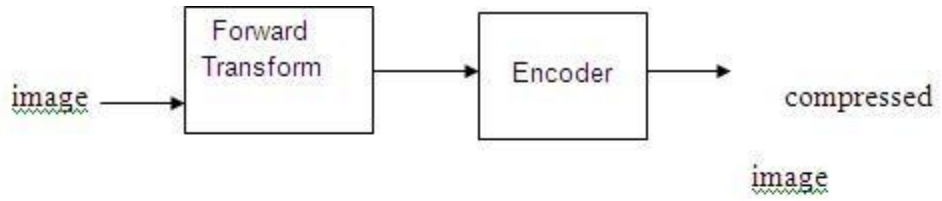
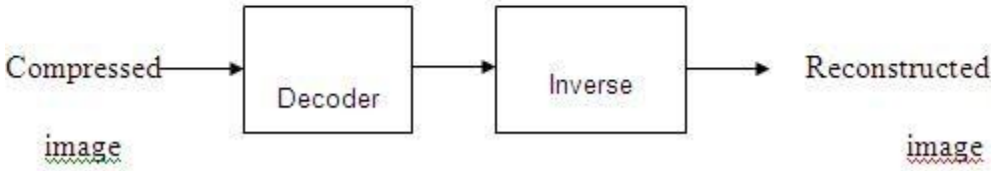


Figure 1.1.a) Block Diagram of Image compression



4

Fig 2.2.11 : Decompression Process for Image

2.3 Image Compression Types:

There are two types' image compression techniques.

1. Lossy Image compression
2. Lossless Image compression
3. Compression ratio:

2.3.1 Lossy Image Compression:

Lossycompression provides higher levels of data reduction but results in a less than perfect reproduction ofthe original image. It provides a high compression ratio. lossy image compression is useful in applicationssuch as broadcast television, videoconferencing, and facsimile transmission, in which a certain amount of error is an acceptable trade-off for increased compression performance. Originally, PGF was designed to quickly and progressively decode lossy compressed aerial images. A lossy compression mode has been preferred, because in an application like a terrain explorer texture data (e.g., aerial orthophotos) is usuallymid-mapped filtered and therefore lossy mapped onto the terrain surface. In addition, decoding lossy compressed images is usually faster than decoding lossless compressed images. In the next test series we evaluate the lossy compression efficiency of PGF. One of the best competitors in this area is for sure JPEG 2000. Since JPEG 2000 has two different filters, we used the one with the better trade-off between compression efficiency and runtime. On our machine the 5/3 filter set has a better trade-off than the other. However, JPEG 2000 has in both cases a remarkable good compression efficiency for very high compression ratios but also a very poor encoding and decoding speed. The other competitor is JPEG. JPEG is one of the most popular image file formats.

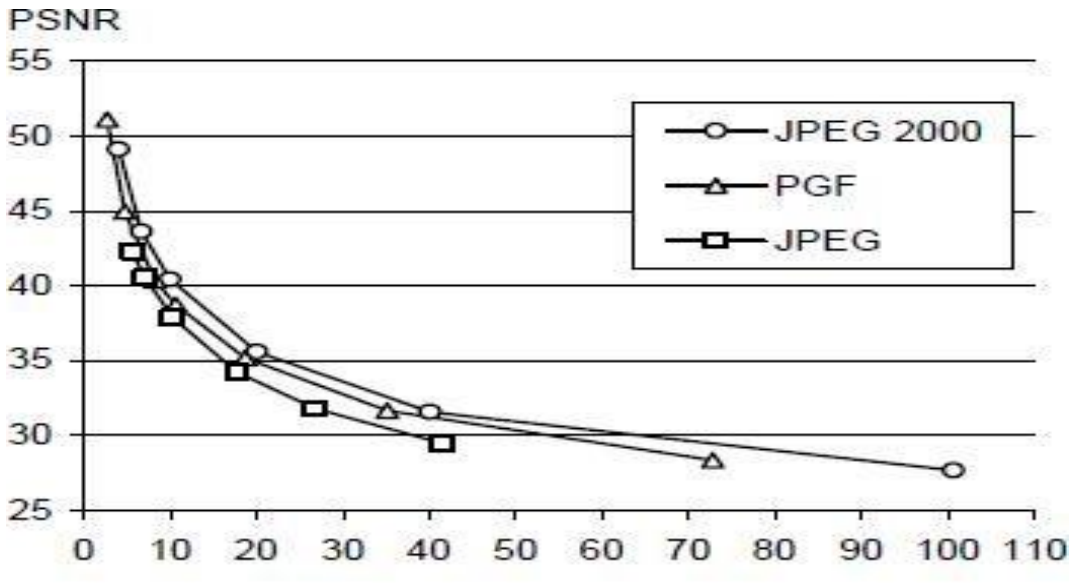


Fig. 4: PSNR of lossy compression in relation to compression ratio.

It is very fast and has a reasonably good compression efficiency for a wide range of compression ratios. The drawbacks of JPEG are the missing losslesscompression and the often missing progressive decoding. Fig. 4 depicts the average rate- distortion behavior for the images in the Kodak test set when fixed (i.e., nonprogressive) lossy compression is used. The PSNR of PGF is on average 3% smaller than the PSNR of JPEG 2000, but 3% better than JPEG. These results are also qualitative valid for our PGF test set and they are characteristic for aerial ortho- photos and natural images. Because of the design of PGF we already know that PGF does not reach the compression efficiency of JPEG 2000. However, we are interested in the trade-off between compression efficiency and runtime. To report this trade-off we show in Table 4 a comparison between JPEG 2000and PGF and in Fig. 5 (on page 8) we show for the same test series as in Fig. 4 the corresponding average decoding times in relation to compression ratios.Table 4 contains for seven different compression ratios (mean values over the compression ratios of the eight images of the Kodak test set) the corresponding average encoding and decoding times in relation to the average PSNR values. In

	JPEG 2000 5/3			PGF		
Ratio	enc	dec	PSNR	enc	dec	PSNR
2.7	1.86	1.35	64.07	0.34	0.27	51.10
4.8	1.75	1.14	47.08	0.27	0.21	44.95
8.3	1.68	1.02	41.98	0.22	0.18	40.39
10.7	1.68	0.98	39.95	0.14	0.13	38.73
18.7	1.61	0.92	36.05	0.12	0.11	35.18
35.1	1.57	0.87	32.26	0.10	0.09	31.67
72.9	1.54	0.85	28.86	0.08	0.08	28.37

Table 4: Trade-off between quality and speed for the Kodak test set

In Fig. 5 we see that the price we pay in PGF for the 3% more PSNR than JPEG is low: for small compression ratios (< 9) decoding in PGF takes two times longer than JPEG and for highercompression ratios (> 30) it takes only ten percent longer than JPEG. These test results are characteristic for both naturalimages and aerial ortho- photos. Again, in the third test series we only use the 'Lena' image. We run our lossy
coder with six different quantization parameters and measure the PSNR in relation to the resulting compression ratios. The results (ratio: PSNR) are:

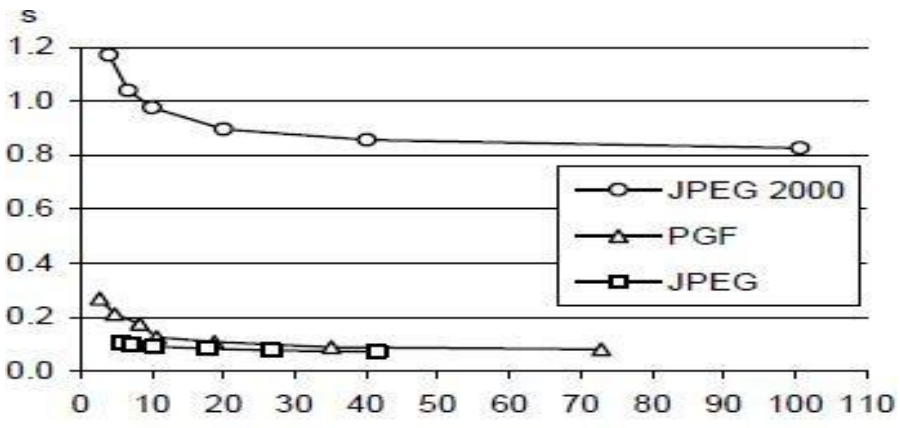


Fig. 5: Decoding time in relation to compression ratio

2.3.2 Lossless Image Compression :

Lossless Image compression is the only acceptable amount of data reduction. It provides a low compression ratio while being lossy. In Lossless Image compression techniques are composed of two relatively independent operations: (1) devising an alternative representation of the image in which its interpixel redundancies are reduced and (2) coding the representation to eliminate coding redundancies. Lossless Image compression is useful in applications such as medical imaginary, business documents and satellite images.Table 2 summarizes the lossless compression efficiency and Table 3 the coding times of the PGF test set. For WinZip we only provide average runtime values, because of missing source code we have to use an interactive testing procedure with runtimes measured by hand. All other values are measured in batch mode.

	WinZip	JPEG-LS	JPEG 2000	PNG	PGF
aerial	1.352	2.073	2.383	1.944	2.314
compound	12.451	6.802	6.068	13.292	4.885
hibiscus	1.816	2.200	2.822	2.087	2.538
houses	1.241	1.518	2.155	1.500	1.965
logo	47.128	16.280	12.959	50.676	10.302
redbrush	2.433	4.041	4.494	3.564	3.931
woman	1.577	1.920	2.564	1.858	2.556
average	**9.71**	**4.98**	**4.78**	**10.70**	**4.07**

Table 2: Lossless compression ratios of the PGF test set.

In Table 2 it can be seen that in almost all cases the best compression ratio

lts show that as far as lossless compression is concerned, PGF performs reasonably well on natural and aerial images. In specific types of images such as ‘compound’ and ‘logo’ PGF is outperformed by far in PNG.

	WinZip		JPEG-LS		JPEG 2000		PNG		PGF	
	enc	dec	enc	dec	enc	dec	enc	dec	enc	dec
a			1.11	0.80	5.31	4.87	3.70	0.19	0.99	0.77
c			1.61	0.38	3.46	3.06	2.95	0.18	0.95	0.80
hi			0.69	0.30	1.45	1.29	1.77	0.10	0.35	0.27
ho			0.65	0.30	1.62	1.47	0.85	0.11	0.41	0.32
l			0.09	0.02	0.26	0.21	0.16	0.01	0.07	0.06
r			0.65	0.44	4.29	4.01	3.61	0.16	0.66	0.59
w			0.39	0.30	1.76	1.63	1.08	0.08	0.35	0.27
av	**1.14**	**0.37**	**0.74**	**0.36**	**2.59**	**2.36**	**2.02**	**0.12**	**0.54**	**0.44**

Table 3: Runtime of lossless compression of the PGF test set

Table 3 shows the encoding (enc) and decoding (dec) times
(measured in seconds) for the same algorithms and images as in Table 2. JPEG 2000 and PGF are both symmetric algorithms, while WinZip, JPEG-LS and in particular PNG are asymmetric with a clearly shorter decoding than encoding time. JPEG 2000, the slowest in encoding and decoding, takes more than four times longer than PGF. This speed gain is due to the simpler coding phase of PGF. JPEG- LS is slightlyslower than PGF during encoding, but slightly faster in decoding images. WinZip and PNG decode even faster than JPEG-LS, but their encoding times are also worse. PGF seems to be the best compromise between encoding and decoding times.

Our PGF test set clearly shows that PGF in lossless mode is best suited fornatural images and aerial orthophotos. PGF is the only algorithm that encodes the three MegaByte large aerial orthophoto in less than a second without a real loss of compression efficiency. For this particular image the efficiency lossis less than
three percent compared to the best. These results should be underlined with our second test set, the Kodaktest set.

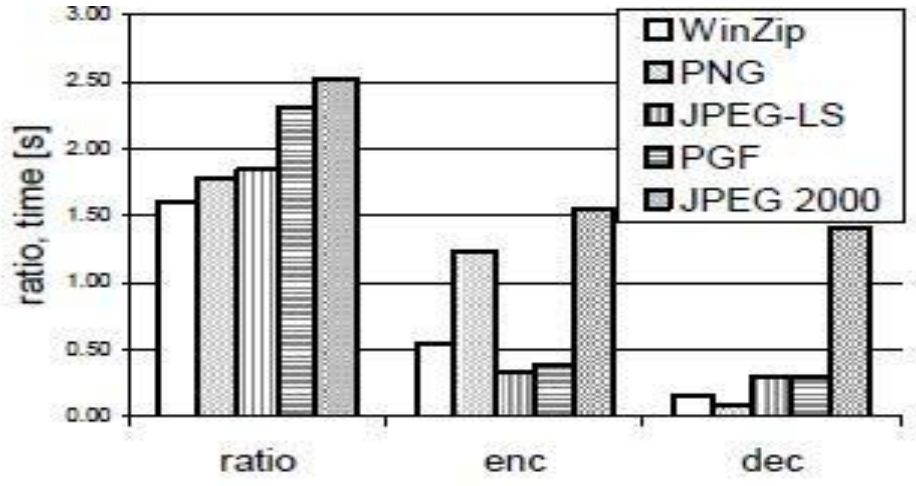


Fig. 3: Lossless compression results of the Kodak test set.

Fig. 3 shows the averages of the compression ratios (ratio), encoding (enc), and decoding (dec) times over all eight images. JPEG 2000 shows in this test set the best compression efficiency followed by PGF,JPEG-LS, PNG, and WinZip. On average PGF is eight percent worse than JPEG 2000. The fact that JPEG2000 has a better lossless compression ratio than PGF does not surprise, because JPEG 2000 is more quality driven than PGF.

However, it is remarkable that PGF is clearly better than JPEG-LS (+21%) and PNG (+23%) for natural images. JPEG-LS shows in the Kodak test set also a symmetric encoding and decoding time behavior. The encoding and decoding times are almost equal to PGF. Only PNG and WinZip can decode faster thanPGF, but they also take longer than PGF to encode.

If both compression efficiency and runtime is important, then PGF is clearly the best of the tested algorithms for lossless compression of natural images and aerial

orthophotos. In the third test we perform our lossless coder on the ‘Lena’ image.

To digitally process an image, it is first necessary to reduce the image to a series of numbers that can be manipulated by the computer. Each number representing the brightness value of the image at a particular location is called a picture element, or pixel. A typical digitized image may have 512 × 512 or roughly 250,000 pixels, although much larger images are becoming common. Once the image has been digitized, there are three basic operations that can be performed on it in the computer. For a point operation, a pixelvalue in the output image depends on a single pixel value in the input image. For local operations, severalneighboring pixels in the input image determine the value of an output image pixel. In a global operation, all of the input image pixels contribute to an output image pixel value.

Correspondingly, these combinations attempt to strike a winning tradeoff: be flexible and hence bring tolerance toward intraclass variation, while also being discriminative enough to be robust to background clutter and intra class similarity. An important feature of our contour-based recognition approach is that it affordsus substantial flexibility to incorporate additional image information. Specifically, we extend the contour-based recognition method and propose a new hybrid recognition method which exploits shape tokens and SIFT features as recognition cues. Shape- tokens and SIFT features are largely orthogonal, where the former corresponds to shape boundaries and the latter to sparse salient image patches. Here, each learned combination can comprise features that are either 1) purely shape-tokens,
purely SIFT features, or 3) a mixture of shape-tokens and SIFT features. The number and types of featuresto be combined together are learned automatically from training images, and represent the more discriminative ones based on the training set. Consequently, by imparting these two degrees of variability(in both the number and the types of features) to a combination, we empower it with even
greater flexibility and discriminative potential. A shorter version of this paper appeared in [9].

2.4 Classification Of Images:

There are 3 types of images used in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Color Image

2.4.1 Binary Image:

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1).This name black and white, monochrome or monochromatic are often used for this concept, but mayalso designate any images that have only one sample per pixel, such as grayscale images Binary imagesoften arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering. Some input/output devices, such as laser printers, fax machines, and bi- level computer displays, can only handle bi-level images

2.4.2 Gray Scale Image:

A grayscale Image is digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray(0-255), varyingfrom black(0) at the weakest intensity to white(255) at the strongest. Grayscale images are distinct from one-bit black-and-white images, which in the context of computerimaging are images with only the two colors, black, and white (also called bi-level or binary images).Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from afull color image; see the section about converting to grayscale.

2.4.3 Color Image:

A (digital) color image is a digital image that includes color information for each pixel. Each pixel has a particular value which determines its appearing color. This value is qualified by three numbers giving the decomposition of the color in the three primary colors Red, Green and Blue. Any color visible to the human eye can be represented this way. The decomposition of a color in the three primary colors is quantified by a number between 0 and 255. For example, white will be coded as R= 255, G = 255, B = 255; black will be known as (R,G,B) = (0,0,0); and say, bright pink will be : (255,0,255).In other words, an image is an enormous two-dimensional array of color values, pixels, each of them codedon 3 bytes, representing the three primary colors. This allows the image to contain a total of 256x256x256 = 16.8 million different colors. This technique is also known as RGB
encoding, and is specifically adapted to human vision

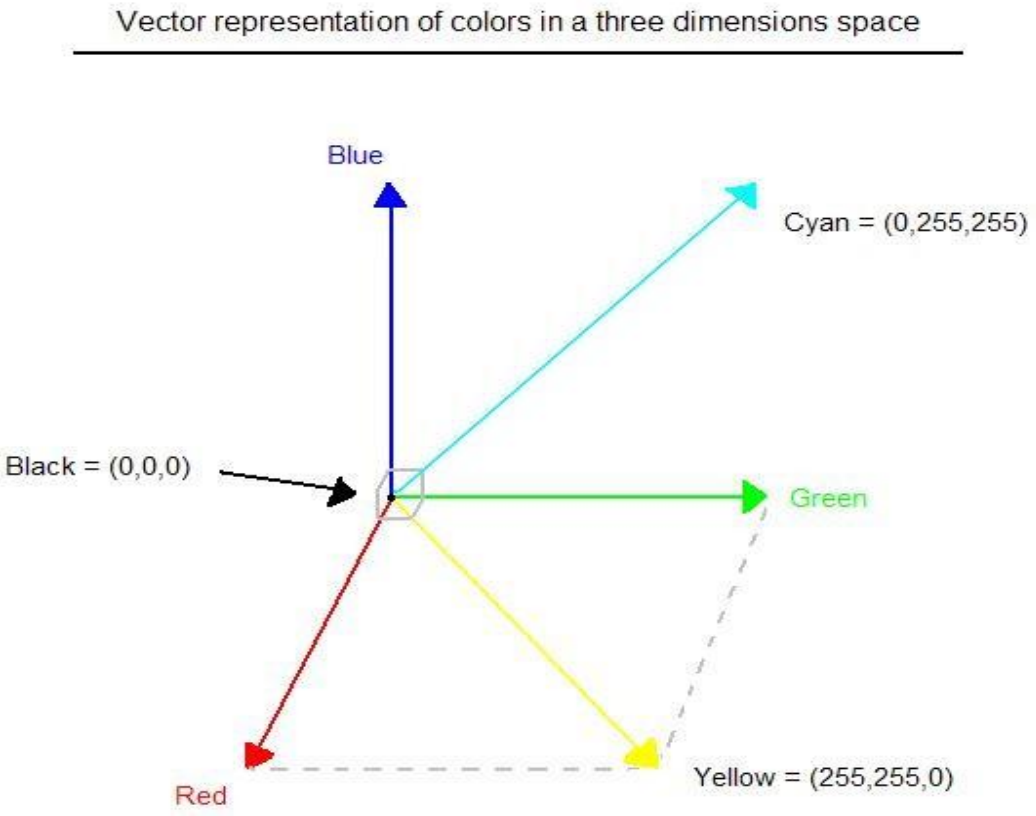


Fig.2.4.1 Hue Saturation Process of RGB SCALE Image

From the above figure, colors are coded on three bytes representing their decomposition on the three primary colors. It sounds obvious to a mathematician to immediately interpret colors as vectors in a threedimensional space where each axis stands for one of the primary colors. Therefore we will benefit from most of the geometric mathematical concepts to deal with our colors, such as norms, scalar product, projection, rotation or distance.

CHAPTER-3
SYSTEM ANALYSIS

3.1 Existing System:

- Principal Component Analysis
- Region based segmentation
- KNN classifier

Principal Component Analysis

PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD).

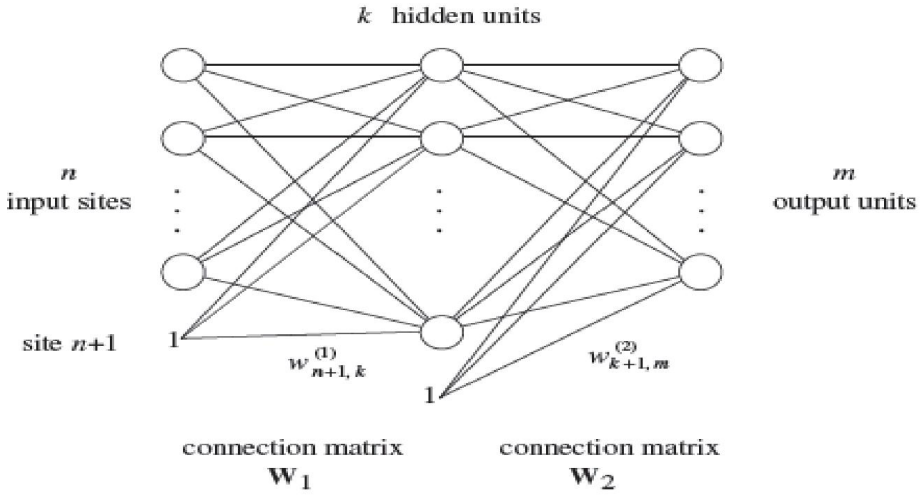
Region growing methods:

The first region growing method was the seeded region growing method. This method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighbouring pixels to the regions. The difference between a pixel's intensity value and the region's mean, δ, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region.

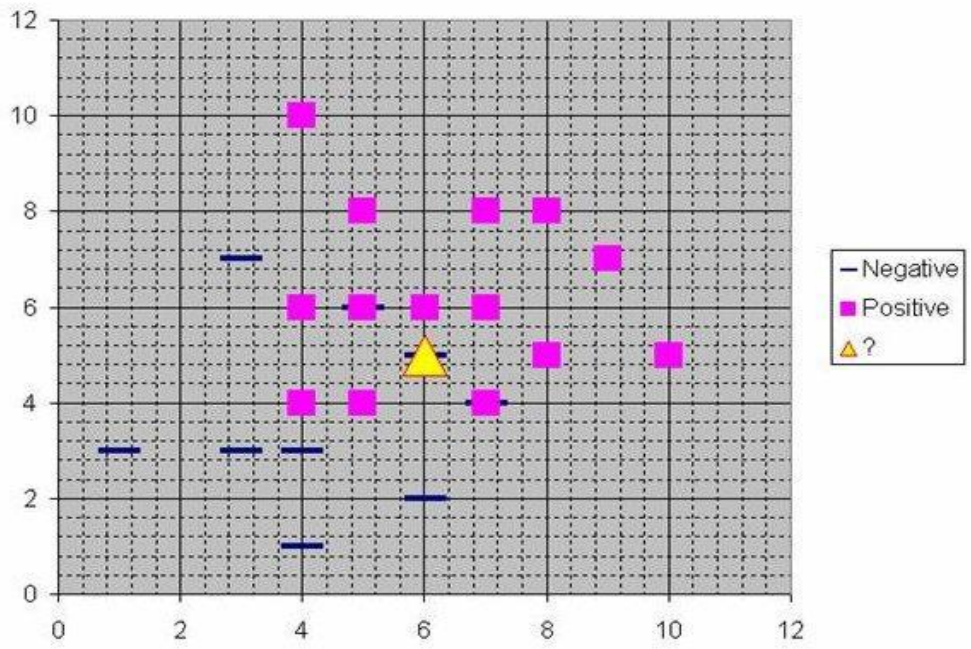
Seeded region growing requires seeds as additional input. The segmentation results are dependent on the choice of seeds. Noise in the image can cause the seeds to be poorly placed. Unseeded region growing is a modified algorithm that doesn't require explicit seeds. It starts off with a single region A_1 – the pixel chosen here

does not significantly influence final segmentation. At each iteration it considers the neighbouring pixels in the same way as seeded region growing. It differs from seeded region growing in that if the minimum δ is less than a predefined threshold T then it is added to the respective region A_j. If not, then the pixel is considered significantly different from all current regions A_i and a new region A_{n+1} is created with this pixel.One variant of this technique, proposed by Haralick and Shapiro (1985), is based on pixel intensities. The mean and scatter of the region and the intensity of the candidate pixel is used to compute a test statistic. If the test statistic is sufficiently small, the pixel is added to the region, and the region's mean and scatter are recomputed. Otherwise, the pixel is rejected, and is used to form a new region.

KNN Classifier:



Although the implementation is very different, back propagation networks are conceptually similar to K-Nearest Neighbor (k-NN) models. The basic idea is that a predicted target value of an item is likely to be about the same as other items that have close values of the predictor variables. Consider this figure:

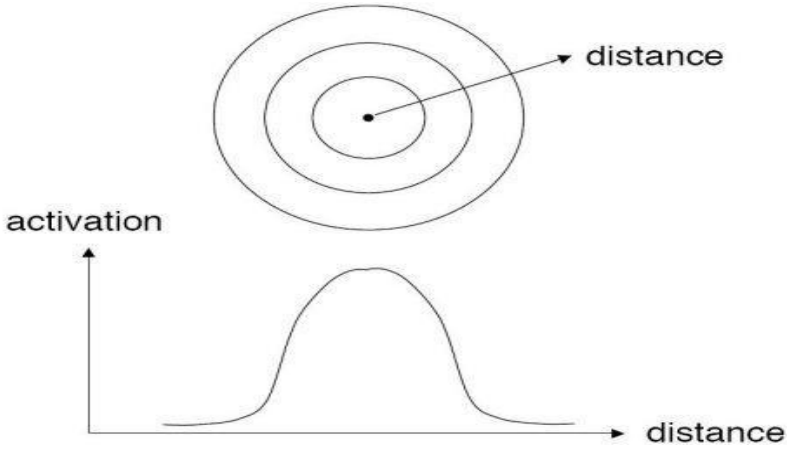


Assume that each case in the training set has two predictor variables, x and y. The cases are plotted using their x,y coordinates as shown in the figure. Also assume that the target variable has two categories, positive which is denoted by a square and negative which is denoted by a dash. Now, suppose we are trying to predict the value of a new case represented by the triangle with predictor values x=6, y=5.1. Should we predict the target as positive or negative.

Notice that the triangle is position almost exactly on top of a dash representing a negative value. But that dash is in a fairly unusual position compared to the other dashes which are clustered below the squares and left of center. So it could be that the underlying negative value is an odd case.

The nearest neighbor classification performed for this example depends on how many neighboring points are considered. If 1-NN is used and only the closest point is considered, then clearly the new point should be classified as negative since it is on top of a known negative point. On the other hand, if 9-NN classification is used and the closest 9 points are considered, then the effect of the surrounding 8 positive points may overbalance the close negative point.

The further some other point is from the new point, the less influence it has.



3.2 Drawbacks:

- High Computational load
- Poor discriminatory power
- Less accuracy in classification

3.3 Proposed System:

- Feature extraction of glcm
- convolution neural network
- Threshold segmentation

3.4 Advantages:

- It is easily identify the disease by using convolution neural network.

3.5 Hardware Requirements:

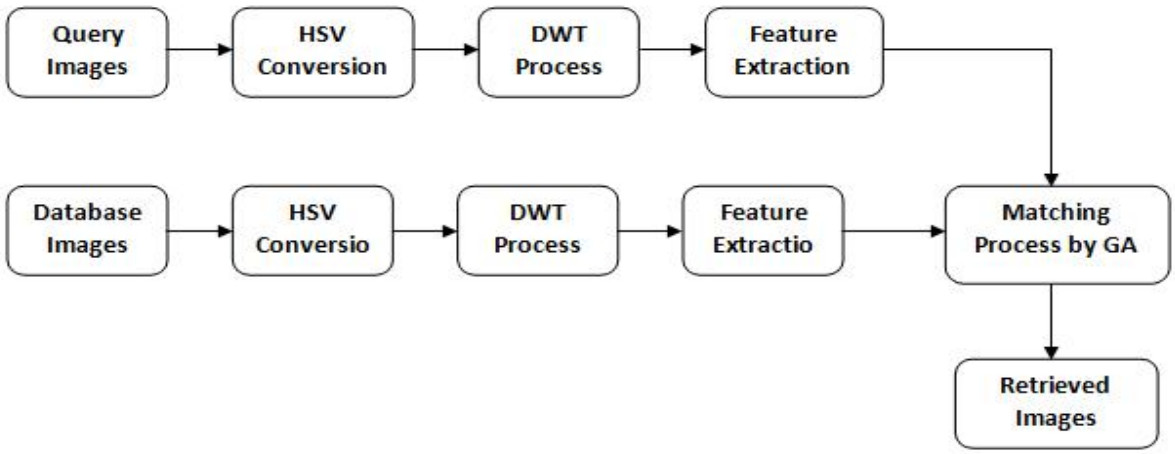
- system
- 4 GB of RAM
- 500 GB of Hard disk

3.6 Software Requirements:

- MATLAB 2018b

CHAPTER-4
SYSTEM DESIGN

4.1 Architecture:



4.2 Modules:

4.2.1 Preprocessing:

Image Pre-processing is a common name for operations with images at the lowest level of abstraction. Its input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. Corruption may come in many forms such as motion blur, noise, and camera misfocus. Image restoration is different from image enhancement in that the latter is designed to emphasize features of the image that make the image more pleasing to the observer, but not necessarily to produce realistic data from a scientific point of view. Image enhancement techniques (like contrast stretching or de-blurring by a nearest neighbor procedure) provided by "Imaging packages" use no a priori model of the process that created the image. With image enhancement noise can be effectively be removed by sacrificing some resolution, but this is not acceptable in many applications. In a Fluorescence Microscope resolution in the z-direction is bad as it is. More advanced image processing techniques must be applied to recover the object. De-Convolution is an example of image restoration method. It is capable of: Increasing resolution, especially in the axial direction removing noise increasing contrast.

4.2.2 Feature Extraction:

Texture analysis

Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface, such as; clouds, leaves, bricks, fabric, etc. It also describes the relationship of the surface to the surrounding environment. In short, it is a feature that describes the distinctive physical composition of a surface.
Texture properties include:

- Coarseness
- Contrast
- Directionality
- Line-likeness
- Regularity
- Roughness

Texture is one of the most important defining features of an image. It is characterized by the spatial distribution of gray levels in a neighborhood [8]. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two-dimensional dependence texture analysis matrix is taken into consideration. This two-dimensional matrix is obtained by decoding the image file; jpeg, bmp, etc.

Methods of Representation

There are three principal approaches used to describe texture; statistical, structural and spectral…

- Statistical techniques characterize textures using the statistical properties of the grey levels of the points/pixels comprising a surface image. Typically, these properties are computed using: the grey level co-occurrence matrix of the surface, or the wavelet transformation of the surface.

- Structural techniques characterize textures as being composed of simple primitive structures called "texels" (or texture elements). These are arranged regularly on a surface according to some surface arrangement rules.
- Spectral techniques are based on properties of the Fourier spectrum and describe global periodicity of the grey levels of a surface by identifying high-energy peaks in the Fourier spectrum [9].

For optimum classification purposes, what concern us are the statistical techniques of characterization… This is because it is these techniques that result in computing texture properties… The most popular statistical representations of texture are:

- Co-occurrence Matrix
- Tamura Texture
- Wavelet Transform

Co-occurrence Matrix

Originally proposed by R.M. Haralick, the co-occurrence matrix representation of texture features explores the grey level spatial dependence of texture [2]. A mathematical definition of the co-occurrence matrix is as follows [4]:

- Given a position operator ***P(i,j)***,
- let ***A*** be an n x n matrix
- whose element ***A[i][j]*** is the number of times that points with grey level (intensity) ***g[i]*** occur, in the position specified by ***P***, relative to points with grey level ***g[j]***.
- Let ***C*** be the ***n*** **x** ***n*** matrix that is produced by dividing ***A*** with the total number of point pairs that satisfy ***P***. ***C[i][j]*** is a measure of the joint probability that a pair of points satisfying ***P*** will have values ***g[i]***, ***g[j]***.
- ***C*** is called a co-occurrence matrix defined by ***P***.

Examples for the operator ***P*** are: "***i*** above ***j***", or "***i*** one position to the right and two below ***j***", etc.

This can also be illustrated as follows… Let t be a translation, then a co-occurrence matrix C_t of a region is defined for every grey-level *(a, b)* by [1]:

$$C_t(a,b) = card\{(s, s+t) \in R^2 \mid A[s] = a, A[s+t] = b\}$$

Here, $C_t(a, b)$ is the number of site-couples, denoted by $(s, s + t)$ that are separated by a translation vector t, with a being the grey-level of s, and b being the grey-level of $s + t$.

For example; with an 8 grey-level image representation and a vector t that considers only one neighbour, we would find [1]:

1	2	1	3	4
2	3	1	2	4
3	3	2	1	1

Figure: Image example

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	0	0	0
2	0	1	0	2	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Figure 4.2.1 : Classical Co-occurrence matrix

At first the co-occurrence matrix is constructed, based on the orientation and distance between image pixels. Then meaningful statistics are extracted from the matrix as the texture representation . Haralick proposed the following texture features :

1. Energy
2. Contrast
3. Correlation
4. Homogenity
5. Entropy

Hence, for each Haralick texture feature, we obtain a co-occurrence matrix. These co-occurrence matrices represent the spatial distribution and the dependence of the grey levels within a local area. Each ***(i,j)*** th entry in the matrices, represents the probability of going from one pixel with a grey level of '***i***' to another with a grey

level of '*j*' under a predefined distance and angle. From these matrices, sets of statistical measures are computed, called feature vectors .

Energy: It is a gray-scale image texture measure of homogeneity changing, reflecting the distribution of image gray-scale uniformity of weight and texture..

$$E = \sum_x \sum_y p(x,y)^2$$

p(x,y) is the GLC M

Contrast: Contrast is the main diagonal near the moment of inertia, which measure the value of the matrix is distributed and images of local changes in number, reflecting the image clarity and texture of shadow depth.

Contrast $I = \sum\sum (x-y)^2 p(x,y)$

Entropy: It measures image texture randomness, when the space co-occurrence matrix for all values are equal, it achieved the minimum value.

$$S = -\sum_x \sum_y p(x,y) \log p(x,y)$$

Correlation Coefficient: Measures the joint probability occurrence of the specified pixel pairs.

Correlation: sum(sum((x- μx)(y-μy)p(x , y)/$\sigma_x\sigma_y$))

Homogeneity: Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

Homogenity = sum(sum(p(x , y)/(1 + [x-y])))

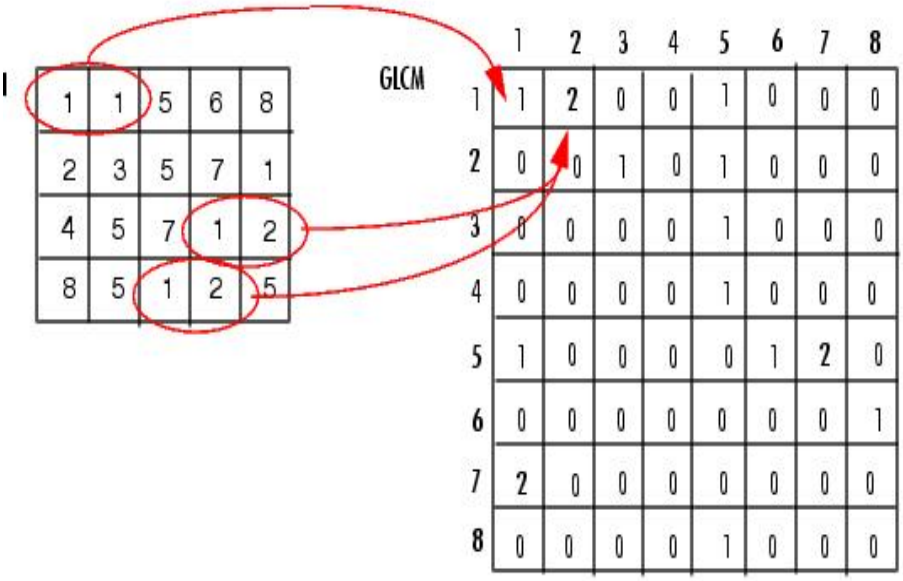
GLCM FEATUES

To create a GLCM, use the graycomatrix function. The graycomatrix function creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value *i* occurs in a specific spatial relationship to a pixel with the value *j*. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right

(horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (*i,j*) in the resultant GLCM is simply the sum of the number of times that the pixel with value *i* occurred in the specified spatial relationship to a pixel with value *j* in the input image. Because the processing required to calculate a GLCM for the full dynamic range of an image is prohibitive, graycomatrix scales the input image. By default, graycomatrix uses scaling to reduce the number of intensity values in gray scale image from 256 to eight. The number of gray levels determines the size of the GLCM. To control the number of gray levels in the GLCM and the scaling of intensity values, using the Num Levels and the Gray Limits parameters of the graycomatrix function. See the graycomatrix reference page for more information.

The gray-level co-occurrence matrix can reveal certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the GLCM are concentrated along the diagonal, the texture is coarse with respect to the specified offset. To illustrate, the following figure shows how graycomatrix calculates the first three values in a GLCM. In the output GLCM, element (1,1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively.

GLCM(1,2)contains the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2. Element (1,3) in the GLCM has the value 0 because there are no instances of two horizontally adjacent pixels with the values 1 and 3. graycomatrix continues processing the input image, scanning the image for other pixel pairs (*i,j*) and recording the sums in the corresponding elements of the GLCM.



To create multiple GLCMs, specify an array of offsets to the graycomatrix function. These offsets define pixel relationships of varying direction and distance. For example, you can define an array of offsets that specify four directions (horizontal, vertical, and two diagonals) and four distances. In this case, the input image is represented by 16 GLCMs. When you calculate statistics from these GLCMs, you can take the average.

You specify these offsets as a *p*-by-2 array of integers. Each row in the array is a two-element vector, [row_offset, col_offset], that specifies one offset. Row_offset is the number of rows between the pixel of interest and its neighbour. Col_offset is the number of columns between the pixel of interest and its neighbour. This example creates an offset that specifies four directions and 4 distances for each direction. After you create the GLCMs, you can derive several statistics from them using the graycoprops function. These statistics provide information about the texture of an image. Statistic such a as Contras, Correlation, Energy, Homogeneity gives information about image.

4.2.3 Neural Network:

Neural networks are predictive models loosely based on the action of biological neurons.

The selection of the name “neural network” was one of the great PR successes of the Twentieth Century. It certainly sounds more exciting than a technical description such as “A network of weighted, additive values with

nonlinear transfer functions". However, despite the name, neural networks are far from "thinking machines" or "artificial brains". A typical artificial neural network might have a hundred neurons. In comparison, the human nervous system is believed to have about $3x10^{10}$ neurons. We are still light years from "Data".

The original "Perceptron" model was developed by Frank Rosenblatt in 1958. Rosenblatt's model consisted of three layers, (1) a "retina" that distributed inputs to the second layer, (2) "association units" that combine the inputs with weights and trigger a threshold step function which feeds to the output layer, (3) the output layer which combines the values. Unfortunately, the use of a step function in the neurons made the perceptions difficult or impossible to train. A critical analysis of perceptrons published in 1969 by Marvin Minsky and Seymore Paper pointed out a number of critical weaknesses of perceptrons, and, for a period of time, interest in perceptrons waned.

Interest in neural networks was revived in 1986 when David Rumelhart, Geoffrey Hinton and Ronald Williams published "Learning Internal Representations by Error Propagation". They proposed a multilayer neural network with nonlinear but differentiable transfer functions that avoided the pitfalls of the original perceptron's step functions. They also provided a reasonably effective training algorithm for neural networks.

Types of Neural Networks:

1) Artificial Neural Network
2) Probabilistic Neural Networks
3) General Regression Neural Networks

DTREG implements the most widely used types of neural networks:

a) Multilayer Perceptron Networks (also known as multilayer feed-forward network),

b) Cascade Correlation Neural Networks,

c) Probabilistic Neural Networks (NN)

d) General Regression Neural Networks (GRNN).

Radial Basis Function Networks:

a) Functional Link Networks,

b) Kohonen networks,

c) Gram-Charlier networks,

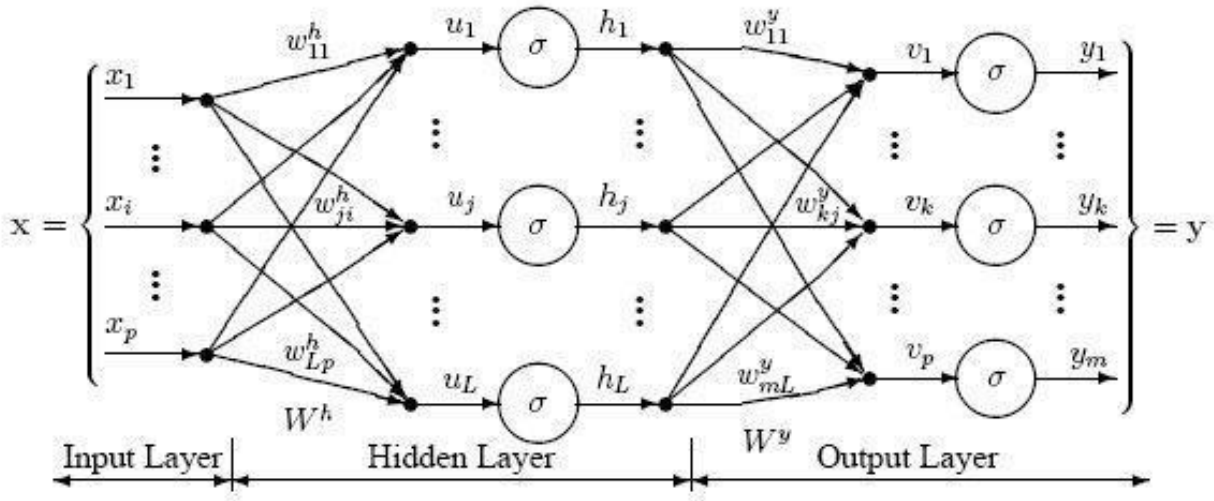
d) Hebb networks,

e) Adaline networks,

f) Hybrid Networks.

The Multilayer Perceptron Neural Network Model

The following diagram illustrates a perceptron network with three layers:



This network has an **input layer** (on the left) with three neurons, one **hidden layer** (in the middle) with three neurons and an **output layer** (on the right) with three neurons.

There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used to represent the *N* categories of the variable.

Input Layer — A vector of predictor variable values ($x_1...x_p$) is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going into the neuron.

Hidden Layer — Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight (w_{ji}), and the resulting weighted values are added together producing a combined value u_j. The weighted sum (u_j) is fed into a transfer function, σ, which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output Layer — Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (w_{kj}), and the resulting weighted values are added together producing a combined value v_j. The weighted sum (v_j) is fed into a transfer function, σ, which outputs a value y_k. The y values are the outputs of the network.

If a regression analysis is being performed with a continuous target variable, then there is a single neuron in the output layer, and it generates a single y value. For classification problems with categorical target variables, there are N neurons in the output layer producing N values, one for each of the N categories of the target variable.

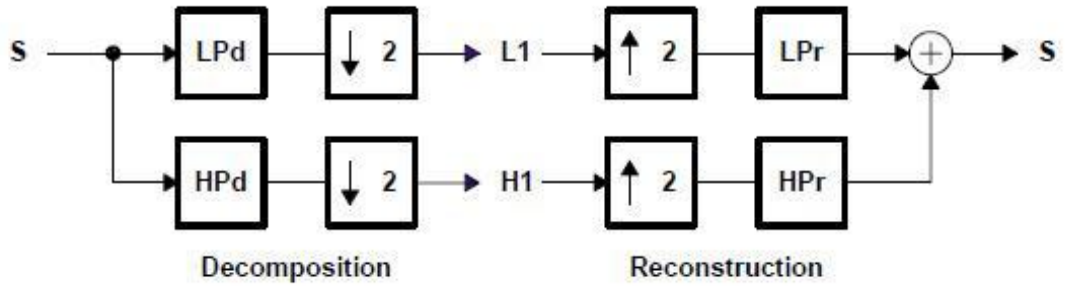
Neural Networks (NN):

Neural Network (NN) and General Regression Neural Networks (GRNN) have similar architectures, but there is a fundamental difference: networks perform classification where the target variable is categorical, whereas general regression neural networks perform regression where the target variable is continuous. If you select a NN/GRNN network, DTREG will automatically select the correct type of network based on the type of target variable.

4.2.4 DWT:

Discrete Wavelet Transform (DWT)

The discrete wavelet remodel (DWT) became superior to use the wavelet rework to the digital international. Filter banks are used to approximate the behavior of the non-prevent wavelet remodel. The sign is decomposed with a immoderate-skip smooth out and a low-bypass clear out. The coefficients of these filters are computed using mathematical evaluation and made to be had to you. See Appendix B for more records about those computations.



Where,

LP d: Low Pass Decomposition Filter

HP d: High Pass Decomposition Filter

LP r: Low Pass Reconstruction Filter

HP r: High Pass Reconstruction Filter

The wavelet literature offers the filter coefficients to you in tables. An example is the Daubechies filters for wavelets. These filters rely upon a parameter p called the vanishing 2nd.

Wavelets Image Processing

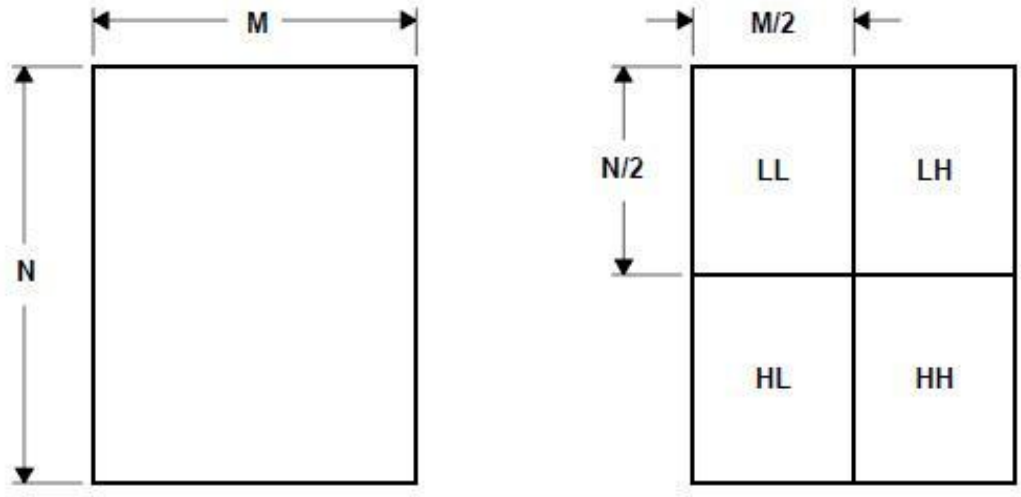
Wavelets have located a massive type of programs within the photo processing discipline. The JPEG 2000 famous uses wavelets for photo compression. Other photo processing programs which incorporates noise reduction, issue detection, and finger print evaluation have additionally been investigated inside the literature.

Wavelet Decomposition of Images

In wavelet decomposing of an photo, the decomposition is achieved row thru row after which column through column. For example, proper here is the technique for an N x M photograph. You filter out every row after which down-sample to acquire N x (M/2) pictures. Then clear out each column and subsample the filter output to attain four (N/2) x (M/2) pictures of the four sub snap shots obtained as visible in Figure 12, the most effective acquired through the use of low-pass filtering the rows and columns is referred to as the LL image.

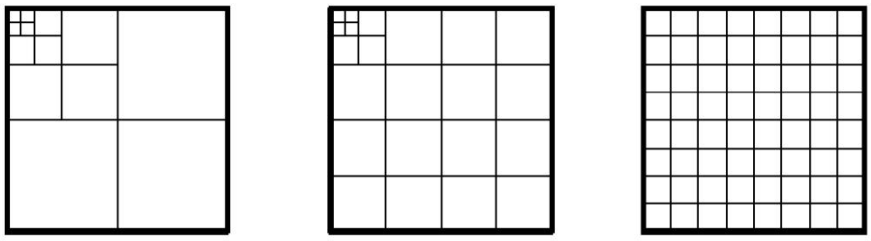
The one acquired with the aid of way of low-skip filtering the rows and high-pass filtering the columns is referred to as the LH snap shots. The one obtained thru immoderate-bypass filtering the rows and espresso-skip filtering the columns is referred to as the HL photo. The sub image received by using manner of the use of immoderate-bypass filtering the rows and columns is referred to as the

HH photograph. Each of the sub snap shots obtained on this fashion can then be filtered and sub sampled to acquire 4 extra sub pictures. This machine can be persisted till the popular sub band form is acquired.



4.2.2 Original Image one level 2-D Decomposition

Three of the maximum well-known strategies to decompose an photograph are: pyramid, spacl, and wavelet packet, as shown in Figure below.



2.4 Three popular wavelet decomposition structures on image

a) pyramid b) space c) wavelet packet

- In the form of pyramid decomposition, most effective the LL subimage is decomposed after each decomposition into four greater subimages.
- In the form of wavelet packet decomposition, every subimage(LL, LH,HL, HH) is decomposed after every decomposition.
- In the shape of spacl, after the primary diploma of decomposition, each subimage is decomposed into smaller subimages, after which only the LL subimage is decomposed.

Figure14 shows a three-diploma decomposition image of pyramid form.

(a) (b)

Figure.4.2.3 a) original image b) three level pyramid structure decomposition.

In the thing I development level, the JPEG 2000 modern-day lets in the pyramid decomposition form. In the destiny all three structures may be supported.

For dimensions, the C55x IMGLIB offers capabilities for pyramid and packet decomposition and reconstruction. Complete data approximately those functions may be determined within the C55x IMGLIB.

2-D discrete wavelet remodel

void IMG_wave_decom_two_dim(short **photo, brief * wksp, int width, int height, int *wavename, int degree);

2-D inverse discrete wavelet rework

void IMG_wave_recon_two_dim(short **photo, short * wksp, int width, int height, int *wavename, int stage);

2-D discrete wavelet bundle rework

void IMG_wavep_decom_two_dim(quick **photo, quick * wksp, int width, int pinnacle, int *wavename, int degree);

2-D inverse discrete wavelet bundle deal redesign

void IMG_wavep_recon_two_dim(short **image, brief * wksp, int width, int top, int *wavename, int degree);

Wavelet transform and multistage analysis

The wavelet transform has been exploited with great success across the gamut of signal processing applications, in the process, often redefining the state of the art performance. In a nutshell, the DWT replaces the infinitely oscillating sinusoidal

basis functions of the Fourier transform with a set of locally oscillating basis functions called *wavelets*. In the classical setting, the wavelets are stretched and shifted versions of a fundamental, real-valued band pass wavelet $\psi(t)$. When carefully chosen and combined with shifts of a real-valued low-pass scaling function $\varphi(t)$, they form an orthonormal basis expansion for one-dimensional (1-D) real-valued continuous-time signals. That is, any finite energy analog signal $x(t)$ can be decomposed in terms of wavelets and scaling functions via

$$x(t) = \sum_{n=-\infty}^{\infty} c(n)\,\phi(t-n) + \sum_{j=0}^{\infty}\sum_{n=-\infty}^{\infty} d(j,n)\,2^{j/2}\,\psi(2^j t - n).$$

The scaling coefficients $c(n)$ and wavelet coefficients $d(j, n)$ are computed via the inner products,

$$c(n) = \int_{-\infty}^{\infty} x(t)\,\phi(t-n)\,dt,$$
$$d(j,n) = 2^{j/2}\int_{-\infty}^{\infty} x(t)\,\psi(2^j t - n)\,dt.$$

They provide a time-frequency analysis of the signal by measuring its frequency content (controlled by the scale factor j) at different times (controlled by the time shift n).There exists a very efficient, linear time complexity algorithm to compute the coefficients $c(n)$ and $d(j, n)$ from a fine-scale representation of the signal (often simply N samples) and vice versa based on two octave-band, discrete-time FBs that recursively apply a discrete-time low-pass filter $h0(n)$, a high-pass filter$h1(n)$, and up sampling and down sampling operations. These filters provide a convenient parameterization for designing wavelets and scaling functions with desirable properties, such as compact time support and fast frequency decay (to ensure the analysis is as local as possible in time frequency) and orthogonality to low-order polynomials (vanishing moments)

This corresponds to a rotation of both filters in the z-plane by 90°. If $h0(n)$ and $h1(n)$ satisfy the PR conditions, then so will $hp(n)$ and $hn(n)$. The given

low-pass/high-pass filters*h*0*(n)*, *h*1*(n)* illustrated in the frequency domain, the complex filters *h*p*(n)*, *h*n*(n)* are illustrated in the frequency domain in Figure 4. When used by itself, this complex can effectively separate the positive and negative frequency components of a signal; in a discrete-time sense, *h*p*(n)* and *h*n*(n)* are approximately analytic.

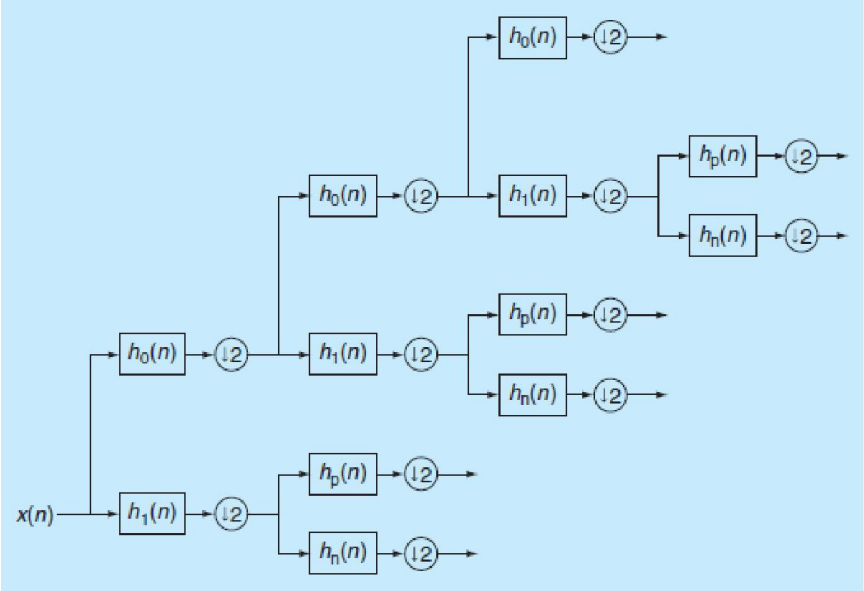


Fig 4.2.4: Analysis FB for the DWT with invertible complex post-filtering.

Shift Variance

A small shift of the signal greatly perturbs the wavelet coefficient oscillation pattern around singularities. Shift variance also complicates wavelet-domain processing; algorithms must be made capable of coping with the wide range of possible wavelet coefficient patterns caused by shifted singularities

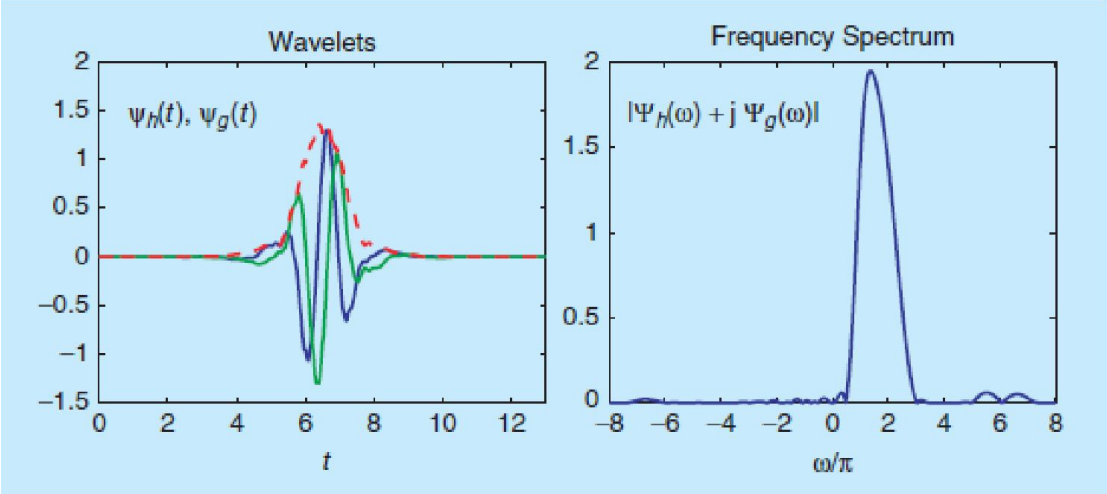


Fig4.2.5: A q-shift complex wavelet corresponding to a set of orthonormal dual- tree filters of length

To better understand wavelet coefficient oscillations and shift variance, consider a piecewise smooth signal *x(t−t0)* like the step function

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

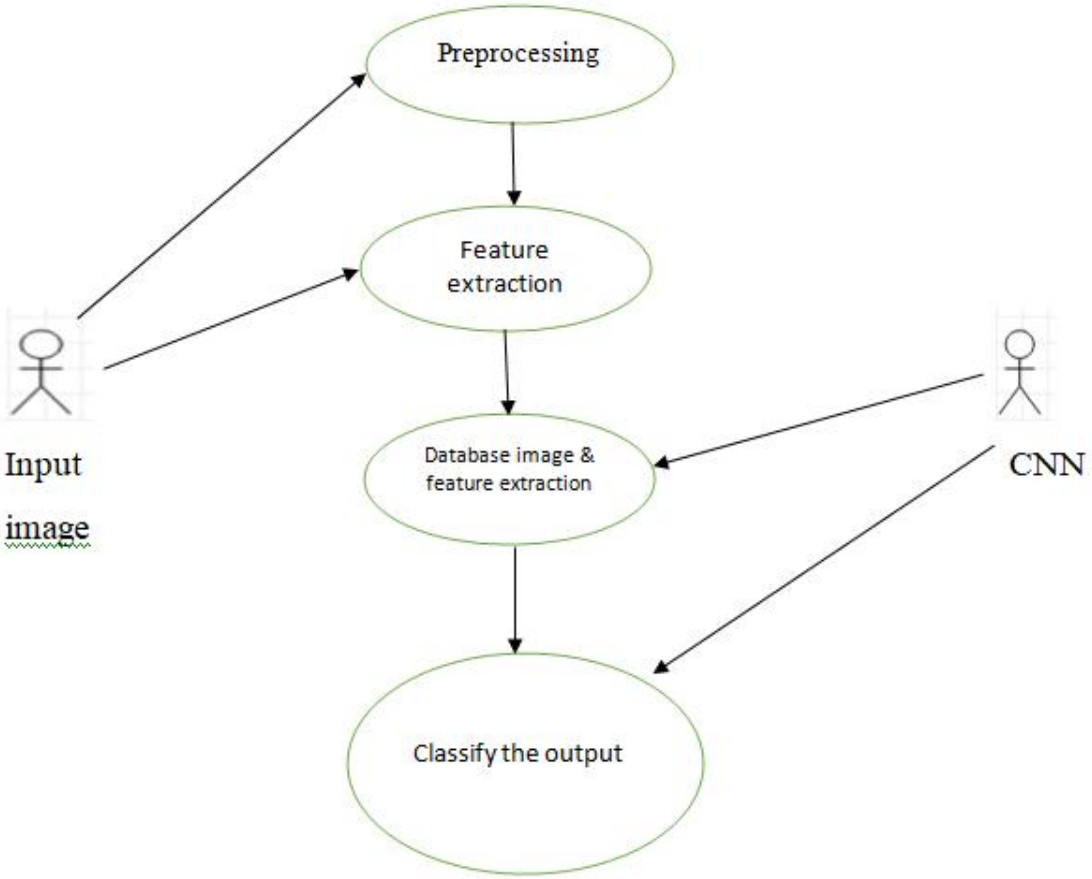
Analyzed by a wavelet basis having a sufficient number of vanishing moments. Its wavelet coefficients consist of samples of the step response of the wavelet

$$d(j, n) \approx 2^{-3j/2} \Delta \int_{-\infty}^{2^j t_0 - n} \psi(t)\, dt,$$

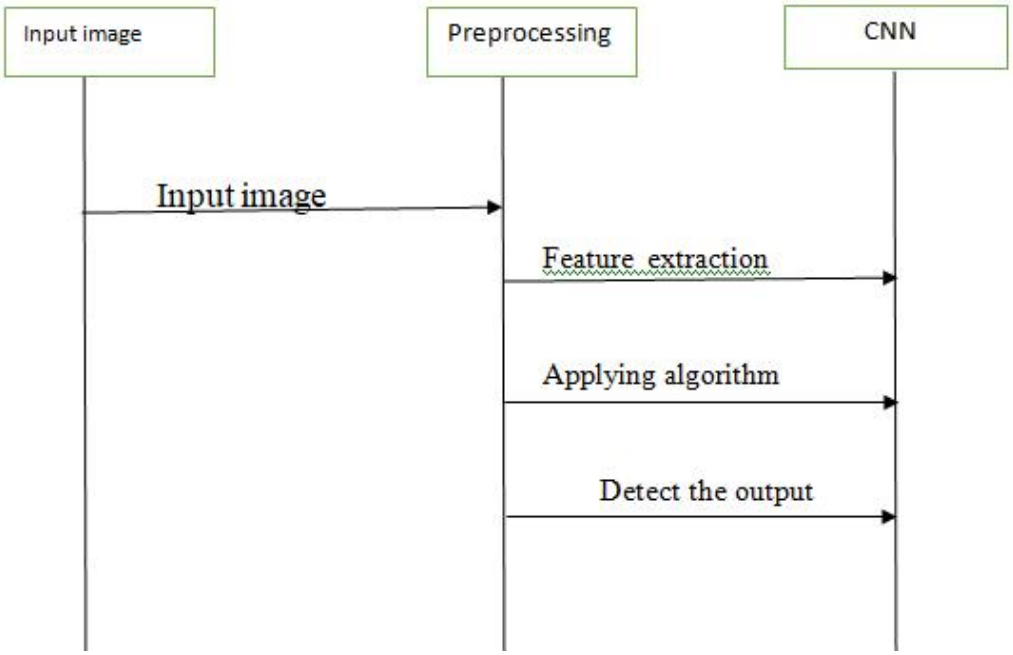
Where is the height of the jump: Since *ψ(t)* is a band pass function that oscillates around zero, so does its step response *d(j, n)* as a function of *n* (recall Figure 1). Moreover, the factor2 *j* in the upper limit (*j*≥ 0) amplifies the sensitivity of *d(j, n)*to the time shift *t*0, leading to strong shift variance.

4.3 UML:

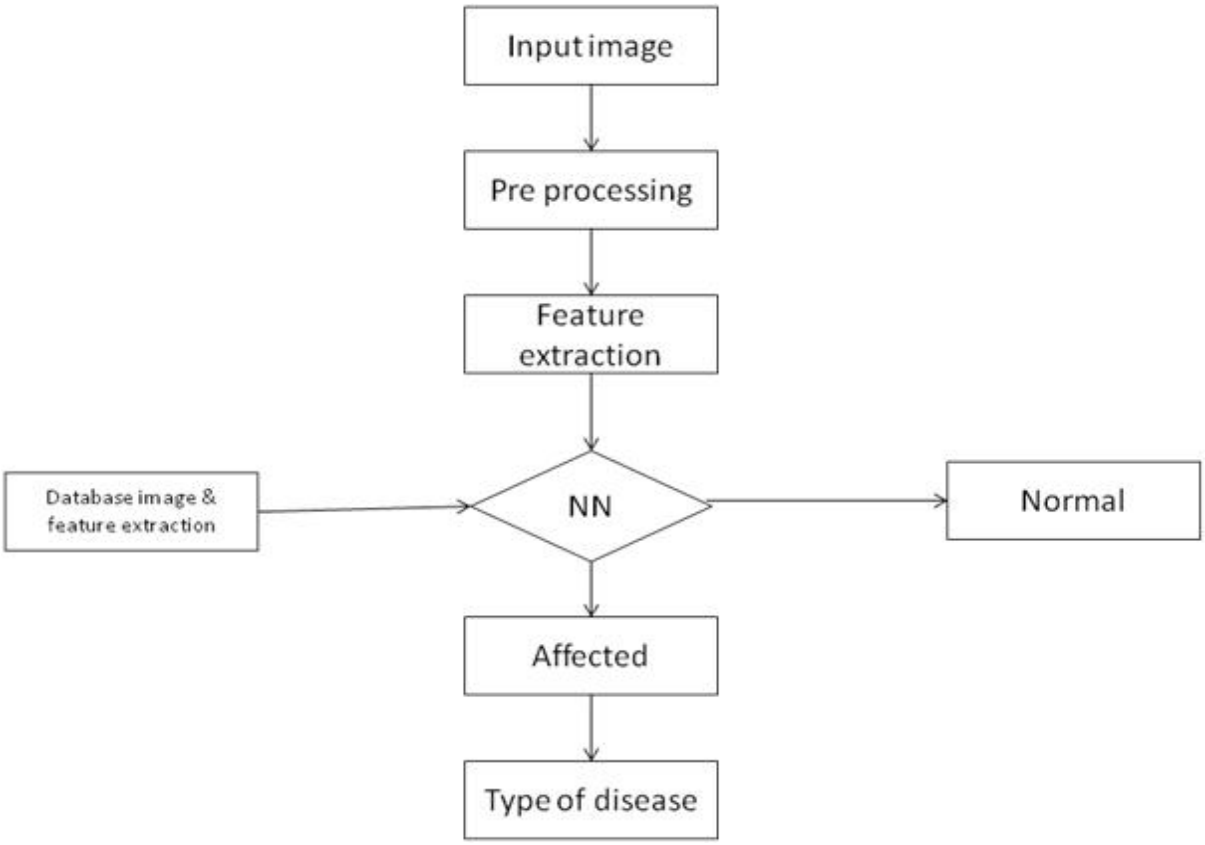
4.3.1 USECASE DIAGRAM:



4.3.2 SEQUENCE DIAGRAM:



4.3.3 ACTIVITY DIAGRAM:



CHAPTER-5
IMPLEMENTATION

5.1 MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve

particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB System

The MATLAB system consists of five main parts:

Development Environment. This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function Library. This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

The MATLAB Language. This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

Handle Graphics®. This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API). This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine,

5.2 Source Code:

```
clc;
clear all;
close all;

cd Input
    [filename, pathname] = uigetfile('*.jpg;*.bmp', 'Pick an Image
file');
if isequal(filename,0) || isequal(pathname,0)
    warndlg('User pressed cancel');
    return;
else
    disp(['User selected ', fullfile(pathname, filename)]);
    inp = imread(filename);
    inp=imresize(inp,[256 256]);
    [r c p] = size(inp);
cd ..
if p==3
    hh = rgb2hsv(inp);
else
    hh = im;
end
figure;
imshow(inp);title('Test Image');
figure;
imshow(hh,[]);title('HSV Image');
end
%%% Plane Separation %%%
Rch = inp(:,:,1);
Gch = inp(:,:,2);
Bch = inp(:,:,3);
```

```
figure;
subplot(1,3,1); imshow(Rch);title('red channel');
subplot(1,3,2); imshow(Gch);title('green channel');
subplot(1,3,3); imshow(Bch);title('blue channel');
inp22=im2bw(inp);
figure;
imshow(inp22);
title('Segmentation Image');

inp22=double(inp22);

boundary = bwboundaries(inp22);
figure;
imtool(inp); title('Abnormal  Localization');
hold on;
 for ii=1:1:length(boundary)
        btemp = boundary{ii};
        plot(btemp(:,2),btemp(:,1),'r','LineWidth',2);
   end
hold off;

%%% Feature Extraction %%%
%%% Color Features %%%
h = hh(:,:,1);
s = hh(:,:,2);
h = double(h);s = double(s);

%%%%%% Mean and Covariance features
HMn = mean(mean(h));
SMn = mean(mean(s));
HCv = sum(sum(cov(h)))./length(h).^2;
```

```
SCv = sum(sum(cov(s))./length(s).^2;
F1 = [HMn SMn HCv SCv];

%%% Texture Features %%%
gim = rgb2gray(inp);
featt = w_feat(gim);
F2 = featt';

%%% Gemotrical features
%%% segmentation %%%
[AA1, AA2, AA3, AA4] = Lclustering(gim);
AA3 = bwfill(AA3,'holes');

figure;
subplot(2,2,1);imshow(AA1,[]);title('1st Cluster');

subplot(2,2,2);imshow(AA2,[]);title('2nd Cluster');

subplot(2,2,3);imshow(AA3,[]);title('3rd Cluster');

subplot(2,2,4);imshow(AA4,[]);title('4th Cluster');

[Sout,Cnt] = bwlabel(AA3,8);
Gprops = regionprops(Sout,'all');

for Ci=1:1:Cnt
    Garea(Ci) = Gprops(Ci).Area;
end
[Amax,Aind] = max(Garea);

Gfeat(1) = Gprops(Aind).Area;
Gfeat(2) = Gprops(Aind).EulerNumber;
```

```
Gfeat(3) = Gprops(Aind).Perimeter;
Gfeat(4) = Gprops(Aind).Eccentricity;
Gfeat(5) = Gprops(Aind).Orientation;
Gfeat(6) = Gprops(Aind).EquivDiameter;
Btemp = Gprops(Aind).BoundingBox;
Gfeat(7) = Btemp(3); Gfeat(8) = Btemp(4);
F3 = Gfeat;
Qfeat = [F1 F2 F3]';

load netp;

%%%%%Classification by simulating trained network model

cout = sim(netp,Qfeat);
cout = round(mean2(cout));
fprintf('\n');
if isequal(cout,1)
    msgbox('CERCOSPORA','The Result: ');
    disp('Diseases and its CURE : CERCOSPORA');
    disp('      Half Ounce Copper with 1 Gallon Water');
elseif isequal(cout,2)
    msgbox('CERCOSPORIDIUM PERSONATUM','The Result: ');
    disp('Diseases and its CURE : CERCOSPORIDIUM
PERSONATUM');
    disp('      0.97 Gllitre followed by Triazole');
elseif isequal(cout,3)
    msgbox('PHAEOISARIOPSIS PERSONATA','The Result: ');
    disp('Diseases and its CURE : PHAEOISARIOPSIS PERSONATA');
    disp('      neem leaf extracts (2-5)& for thrice  2weeks & 4
weeks after Planting');
elseif isequal(cout,4)
```

```
    msgbox('ALTERNARIS','The Result: ');
    disp('Diseases and its CURE : ALTERNARIS');
    disp('      Foliar application of Mancozeb 0.3%');
elseif isequal(cout,5)
    msgbox('NORMAL LEAF','The Result: ');
else
    msgbox('DB Updation Required');
 end
```

CHAPTER-6
TESTING

The purpose of testing is to discover errors. Testing is the process of trying 10 discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform hasic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2 Integration Testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3 Functional Testing:

Functional tests provide systematic demonstrations that functions tested are vailable as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be onsidered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.4 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.5 Black Box Testing:

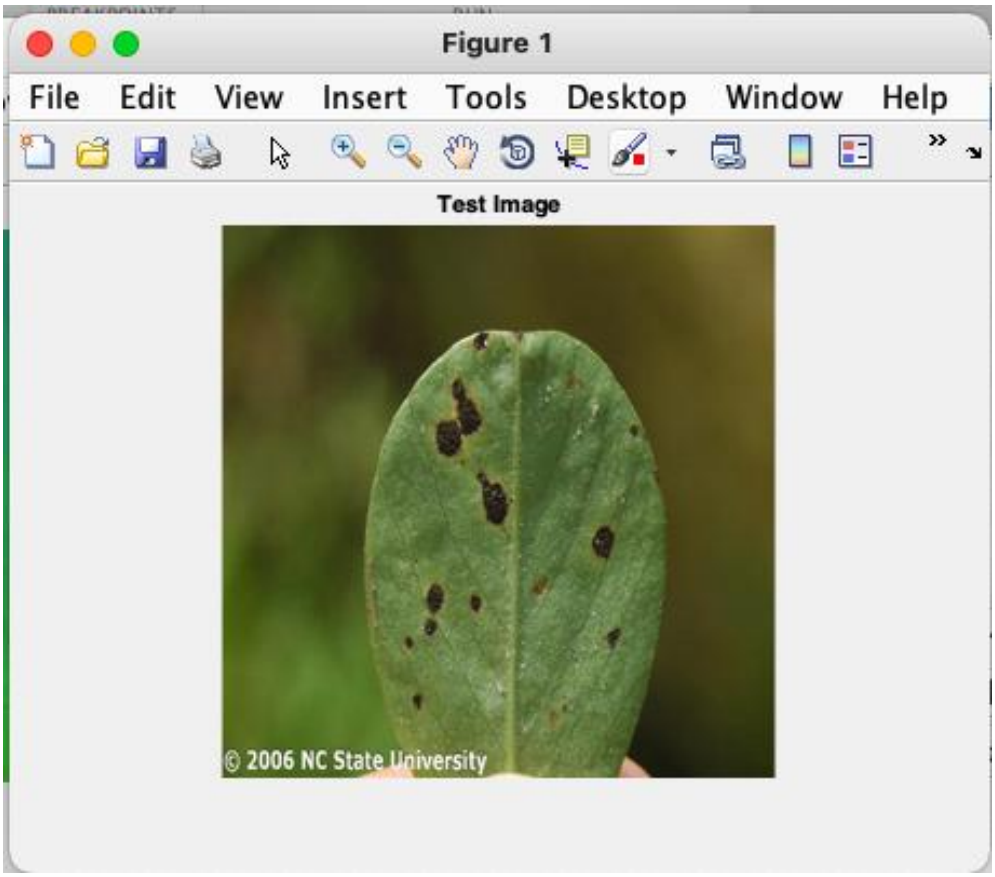
Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements

document, such as specification or requirements document. It is a testing in which the sofiare under test is treated, as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

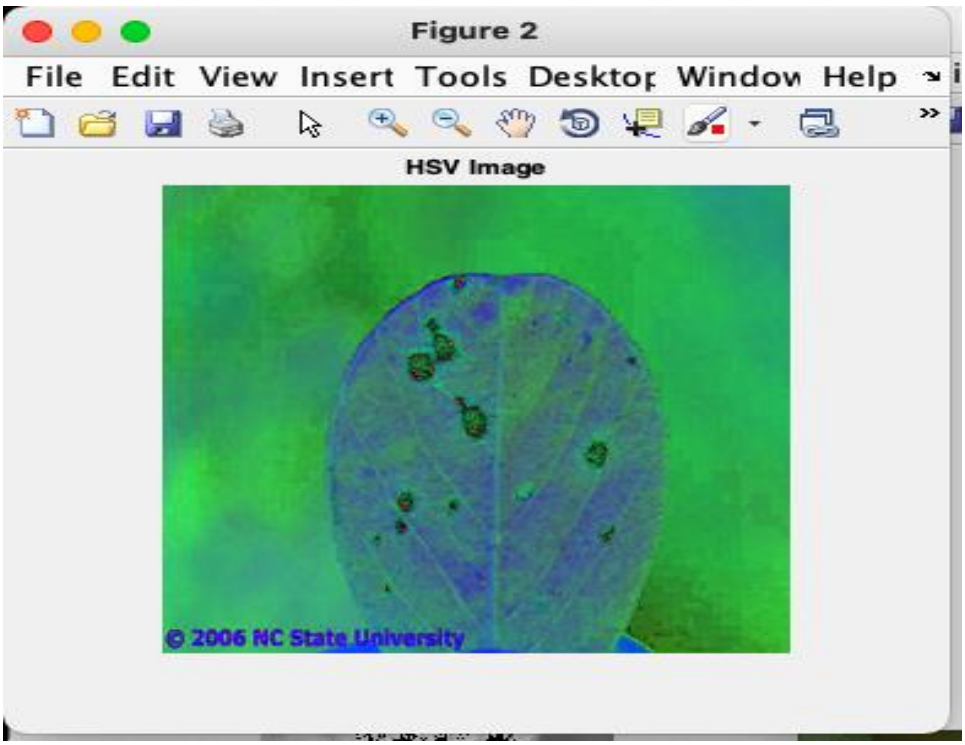
CHAPTER-7
RESULT

7.1 Output:

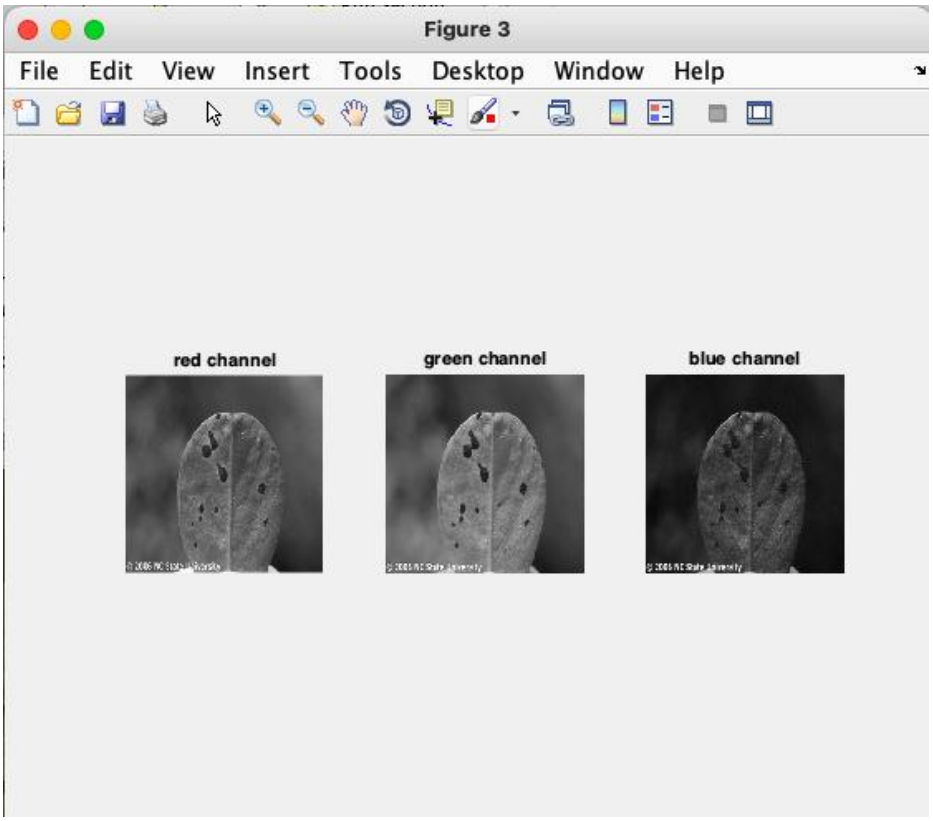
7.3.1 Actual Image:



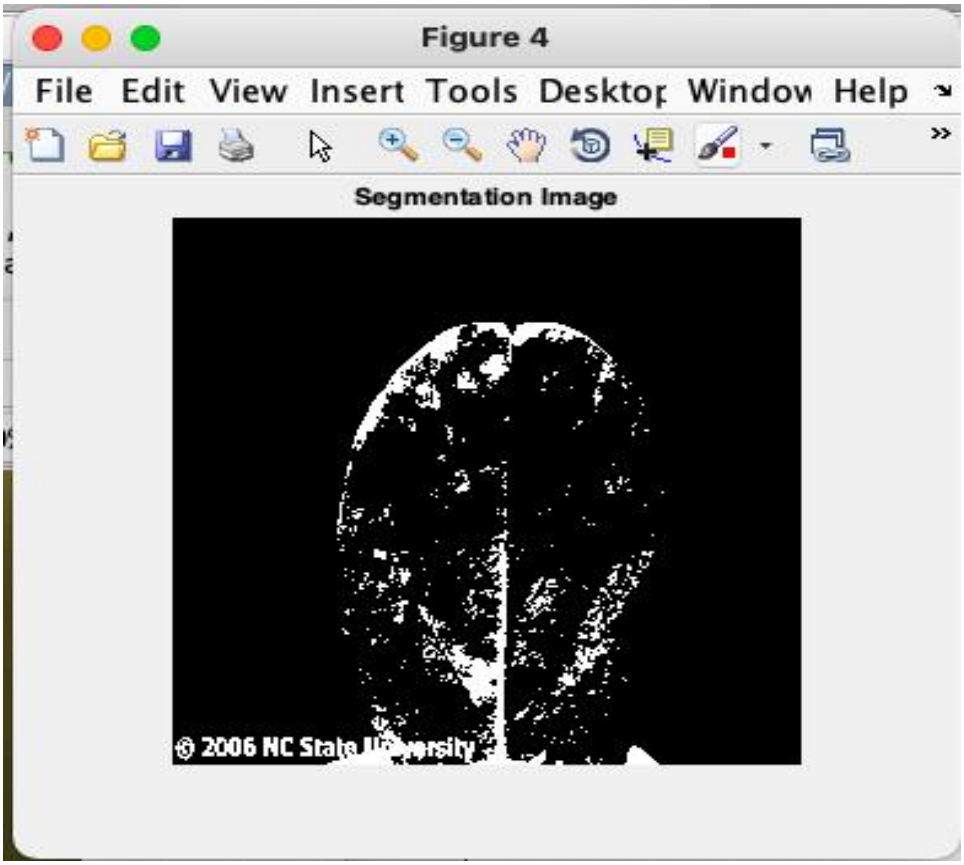
7.3.2 HSV Image:



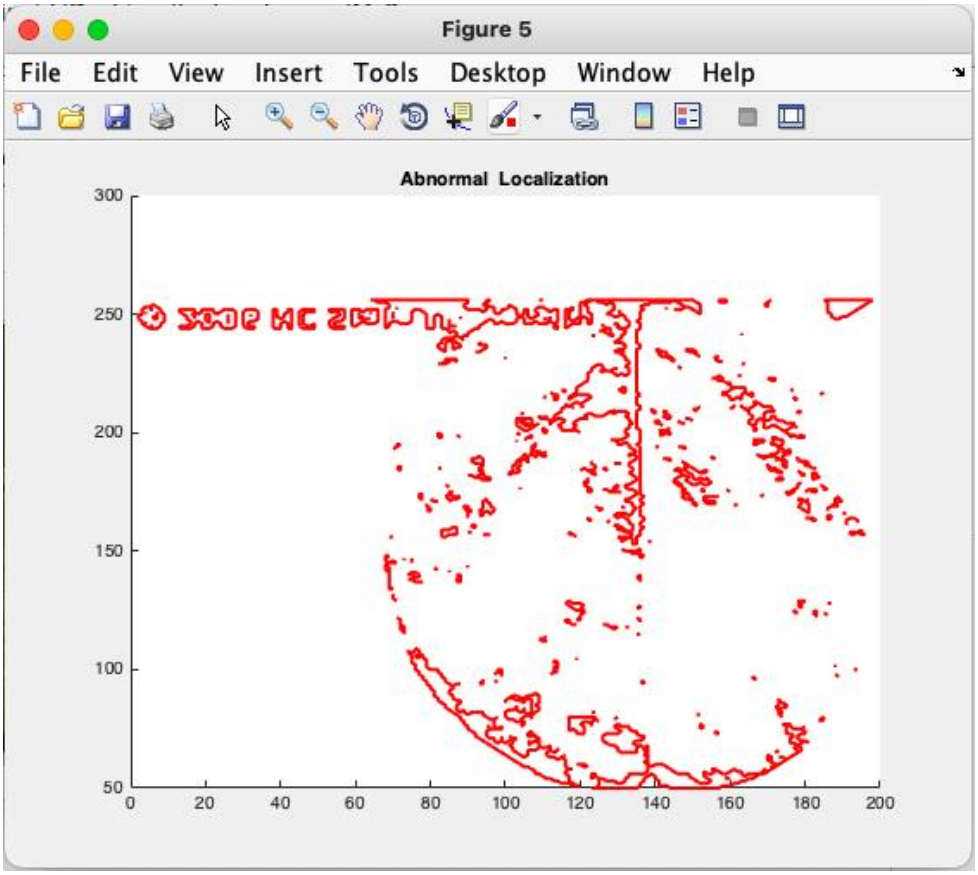
7.3.3 RGB Image:



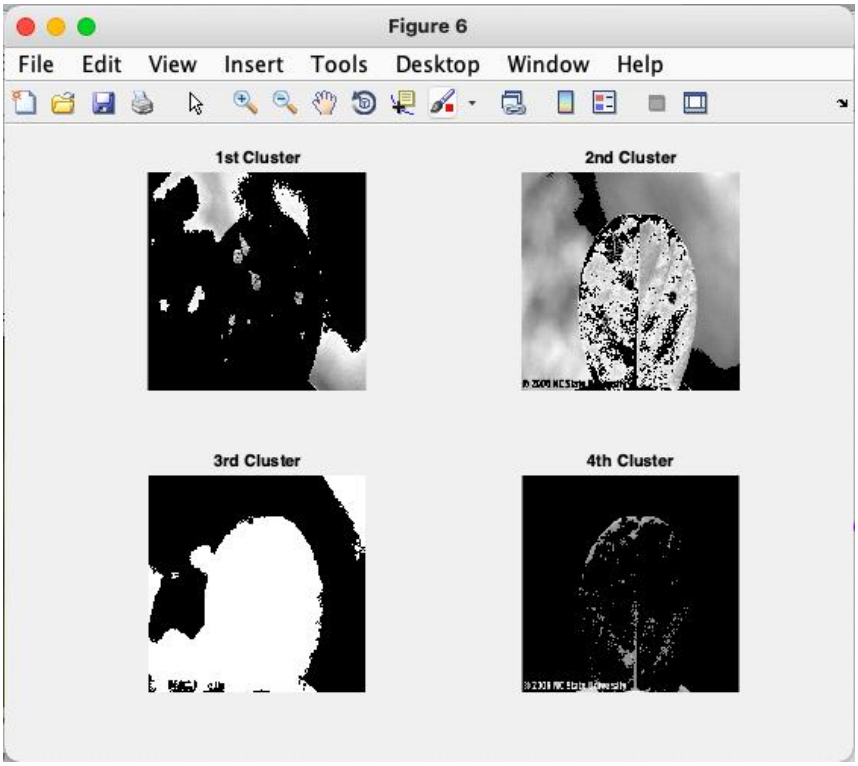
7.3.4 Segmentation Image:



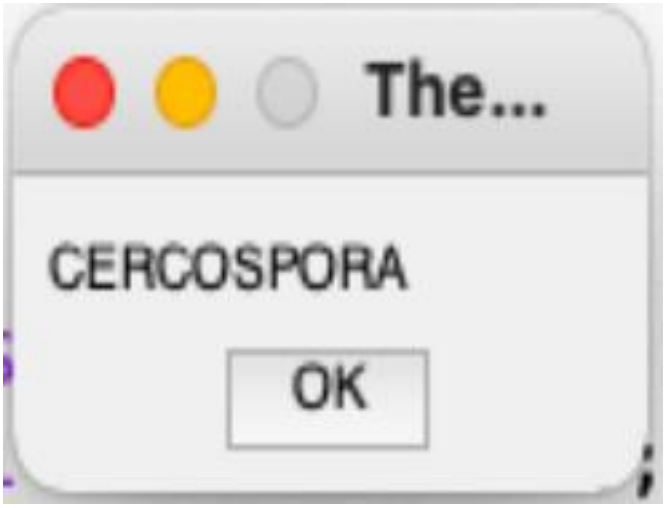
7.3.5 Plot Image:



7.3.6 Cluster Image:



7.3.7 Disease Output:



CHAPTER-8
CONCLUSION

8.1 Conclusion:

The above Literature survey has detailed explanation of the importance of disease detection both to plants and to mankind. To have a meaningful impact of plant diseases & techniques in the area of agriculture, deliberation of proper input is necessary. Research issues addressed here are to develop a systematic approach to detect and recognize the plant diseases would assist farmers and pathologist in prospect exploration. The paper depicts the importance of image processing in agriculture field and considering the type of disease for further research work.

CHAPTER-9

FUTURE ENHANCEMENT

9.1 Future Enhancement:

In future we increased the performance of this process and able to get more accuracy.

REFERENCES

REFERENCES :

[1]Sachin D. Khirade, A.B Patil, "Plant Disease Detection Using Image Processing", International Conference on Computing Communication Control and Automation", 2015.

[2]Prof. Sanjay B. Dhaygude, Mr.Nitin P.Kumbhar, "Agricultural plant Leaf Disease Detection Using Image Processing", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 1, January 2013.

[3]Amandeep Singh ,Maninder Lal Singh, "Automated Color Prediction of Paddy Crop Leaf using Image Processing", International Conference on Technological Innovations in ICT for Agriculture and Rural Development (TIAR 2015), 2015.

[4]M.Malathi, K.Aruli , S.Mohamed Nizar, A.Sagaya Selvaraj, "A Survey on Plant Leaf Disease Detection Using Image Processing Techniques",International Research Journal of Engineering and Technology (IRJET),Volume: 02 Issue: 09, Dec 2015.

[5]MalvikaRanjan, Manasi Rajiv Weginwar, NehaJoshi, Prof.A.B. Ingole, "detection and classification of leaf disease using artificial neural network", International Journal of Technical Research and Applications, 2015.

[6]Y.Sanjana, AshwathSivasamy, SriJayanth, "Plant Disease Detection Using Image Processing Techniques",International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Special Issue 6, May 2015.

[7]Bhumika S.Prajapati, Vipul K.Dabhi Harshadkumar, B.Prajapati, "A Survey on Detection and Classification of Cotton Leaf Diseases", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) –2016.

[8]P.Revathi, M.Hemalatha, "Advance Computing Enrichment Evaluation of Cotton Leaf Spot Disease Detection U sing Image Edge detection", ICCCNT'12.

[9]Mr. Pramod S. landge, Sushil A. Patil, Dhanashree S. Khot, "Automatic Detection and Classification of Plant Disease through Image Processing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, 2013.