CHAPTER-1

INTRODUCTION

1. INTRODUCTION

In India, agriculture is the backbone of economy. 50% of the population is involved in farming activities directly or indirectly. Many varieties of fruits, cereals and vegetables are produced here and exported to other countries. Hence it is necessary to produce high quality products with an optimum yield. As diseases of the plants are unavoidable, detection of plant diseases is essential in the field of Agriculture. In plants, diseases can be found in various parts such as fruits, stems and leaves. The main diseases of plants are viral, fungus and bacterial disease like Alternaria, Anthracnose, bacterial spot, canker, etc.,. The viral disease is due to environmental changes, fungus disease is due to the presence of fungus in the leaf and bacterial disease is due to presence of germs in leaf or plants. The proposed framework can be used to identify leaf diseases. Automatic detection of plant diseases is an important research topic since it is able to automatically detect the diseases from the symptoms that appear on the plant leaves.

Barbedo proposed an automatic method of disease symptoms segmentation in digital photographs of plant leaves, in which color channel manipulation & Boolean operation are applied on binary mask of leaf pixels [1]. He proposed the method of semi-automatic segmentation of plant leaf disease symptoms in which the histograms of the H and color channels are manipulated [2, 3]. Pang et al proposed the method of automatic segmentation of crop leaf spot disease images by integrating local threshold and seeded region growing [4]. Singh and Misra proposed detection of plant leaf diseases using soft computing techniques [5]. Prasad et al proposed unsupervised resolution independent based natural plant leaf disease segmentation approach in which texture based clustering for segmentation is done [6]. Du & Zhang proposed a technique to segment leaf image with non-uniform illumination based on maximum entropy and genetic algorithm (GA) [7]. Dhaygude & Kumbhar proposed agricultural plant leaf disease detection using image processing in which the texture statistics are computed from spatial gray-level dependence matrices (SGDM) [8]. Diao et al reviewed the different methods including edge based, region based, Artificial Neural Network (ANN) etc., for segmentation of plant disease spot [9]. Different methods for automatic leaf image segmentation and disease identification have been proposed in literature [10-14]

1.1 History

A product quality control is fundamentally required in order to gain more value- added product. Many studies show that quality of agricultural products can be reduced from many causes. One of the most important factors of such quality is plant diseases. Consequently, minimizing plant diseases allows substantially improving quality of the products.

Groundnut known as Arachis hypogaea (specific name), is one of the most utilized food plants and widely grown originated in ASIA. Groundnut is an important crop worldwide and over half of the world population relies on it for food. Many people in the world including India eat Groundnut as staple food. However, there are many factors that make groundnut production become slow and less productive. One of the main factors is groundnut disease.

An abnormal condition that injures the plant or leads it to function improperly is called as a disease. Diseases are readily recognized by their symptoms. There are a lot of groundnut disease types which are red disease virus, brown spot disease and many more. Image processing is very beneficial to the agricultural industry. They are more potential and more important to many areas in agricultural technology.

Groundnut Disease Detection System is one of the very beneficial systems. It can help the groundnut farmer detect the disease faster. This study aims to develop a prototype system to automatically detect and classify the groundnut diseases by using image processing technique as an alternative or supplemental to the traditional manual method.

1.2 Objective

The main objective of the project is to detect and classify the groundnut diseases through Image Segmentation Technique. In groundnut leaf the disease cannot be detected at early stages and due to this reasons the whole crop gets damaged. Hence, it results in the decrease of the yield of the crop. In order to prevent the loss of the crop the disease should be detected at early stages.

1.3 Introduction to Digital Image Processing

An image may be defined as a two dimensional function f(x,y), where x and y are spatial coordinates and the amplitude of at any point of coordinates (x,y) is called the intensity or gray level of the image at that point. When x,y and the amplitude values of f are all finite discrete quantities, we call the image is a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location and value. The elements are called pixels.

Vision is most advanced of our sensors, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire Electromagnetic spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors reading where image processing stops and other related areas such as image analysis and computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input and output at a process are images. This is limiting and somewhat artificial boundary. The area of image analysis is in between image processing and computer vision.

There are no clear cut boundaries in the continuum from image processing at one end to complete vision at the other. However on useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, high level processes. Low level process involves primitive operations such as image processing to reduce noise, contrast enhancement and image sharpening.

A low level process is characterized by the fact that both its inputs and outputs are images. Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing and classification of individual objects. A mid level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images.

Finally high level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis and at the far end of the continuum performing the cognitive functions normally associated with human vision. Digital image processing, as already defined as used successfully in a broad range of areas of exceptional social and economic value.

1.3.1 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 and shown in fig 1.3.1.1. 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.

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size 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 1.3.1.1 General image

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size 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 shown in fig 1.3.1.2.

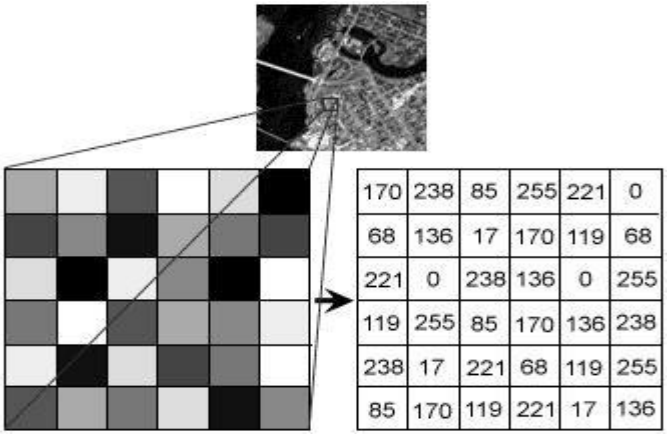
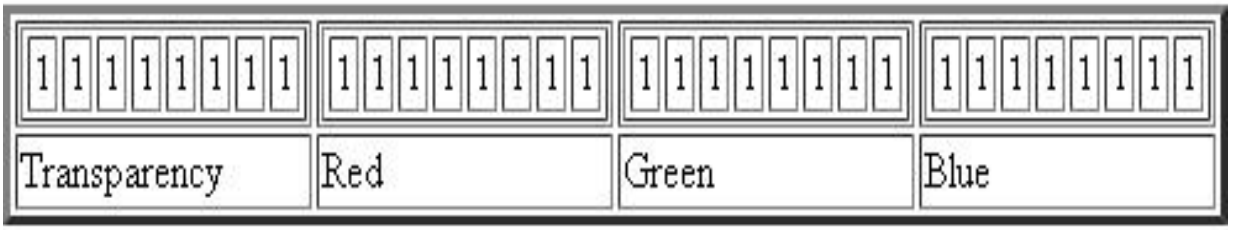


Fig 1.3.1.2 Pixels of an image

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 blueness, and the remaining eight bits the transparency of the pixel.



1.3.2 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 shown following fig1.3.2.1. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

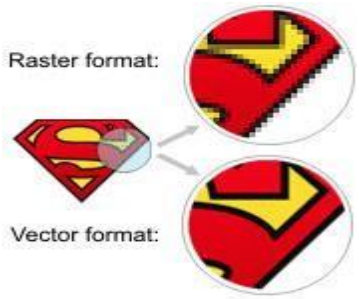


Fig1.3.2.1 Resolution image

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

1.3.3 RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps)

- **JPEG/JFIF:**

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per color (red, green, blue) for a 24-bit total, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format if they will be re-edited, or if small "artifacts" are unacceptable. The JPEG/JFIF format also is used as the image compression algorithm in many Adobe PDF files.

- **EXIF:**

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software.

The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information, etc. When images are viewed or edited by image editing software, all of this image information can be displayed.

- **TIFF:**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively, usually using either the TIFF or TIF filename extension. TIFFs are lossy and lossless. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

- **PNG:**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to the GIF. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas. The lossless PNG format is best suited for editing pictures, and the lousy formats, like JPG, are best for the final distribution of photographic images, because JPG files are smaller than PNG files. PNG, an extensible file format for the lossless, portable, well-compressed storage of raster images. PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is designed to work well in online viewing applications, such as the World Wide Web. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors.

- **GIF:**

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

- **BMP:**

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large. The advantage is their simplicity and wide acceptance in Windows programs.

1.3.4 VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size.

At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics.

- **CGM:**

CGM (Computer Graphics Metafile) is a file format for 2D vector graphics, raster graphics, and text. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations. CGM provides a means of graphics data interchange for computer representation of 2D graphical information independent from any particular application, system, platform, or device.

- **SVG:**

SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need for a versatile, scriptable and all purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip.

1.3.5 Types of Images

A. 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 in the image is the foreground color while the rest of the image is the background color and shown in fig 1.3.5.1.

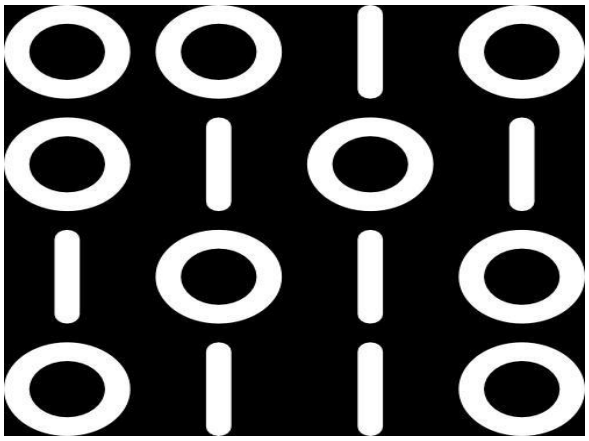


Fig 1.3.5.1 Binary Image

B. Gray scale image

A gray scale image is contains only brightness information. Each pixel value corresponds to a quantity of light. For example an 8-bit image will have a brightness variation form 0 to 255 where 0 represents black and 255 represents white. It measures only light intensity and shown in fig 1.3.5.2.

Fig 1.3.5.2 Gray image

C. Color image

A color image has three values per pixel and they measure the intensity and chrominance of light. Each pixel is a vector of color component. Common color spaces are RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), and CMYK (Cyan, Magenta, Yellow, Black). Colors can be represented with three color components so the typical uncompressed data rate of a color image is three times the data rate of a gray scale image and shown in fig 1.3.1.1. A color image is usually stored in memory as a raster map, a two dimensional array of small integer triplets; or (rarely) as three separate raster maps.

1.4 Fundamental steps in Image Processing

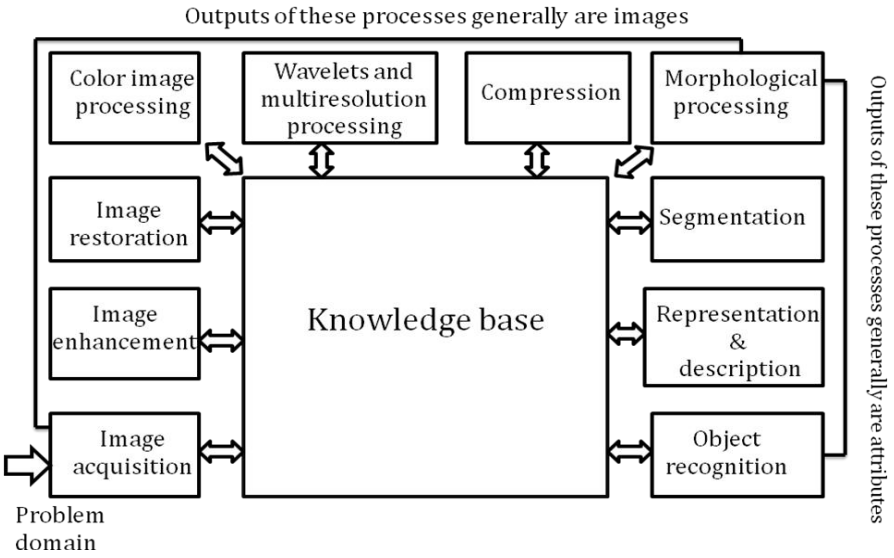


Fig 1.4.1 Fundamental steps in Image Processing

1.4.1 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 monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time shown in below fig 1.4.1.1. In this case, the objects motion past the line.

Fig 1.4.1.1 Digital camera image

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 produces are determined by the application.

Fig 1.4.1.2 digital camera cell

1.4.2 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 interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better” and shown in fig 1.4.2.1. It is important to keep in mind that enhancement is a very subjective area of image processing.

Fig 1.4.2.1 Image enhancement

1.4.3 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 shown in below fig 1.4.3.1.

Fig 1.4.3.1 Image restoration

Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a de-blurring function is considered a restoration technique.

1.4.4 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. This second factor is particularly important in manual image analysis shown in fig 1.4.4.1.

Fig 1.4.4.1 Color & Gray scale image

1.4.5 Wavelets and multi-resolution processing:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late1950's, a more recent transformation, called the wavelet transform, the Fourier transform, whose basis functions are sinusoids, wavelet transforms are base and is now making it even easier to compress, transmit, and analyze many images shown in fig 1.4.5.1. Unlike on small values, called Wavelets, of varying frequency and limited duration.

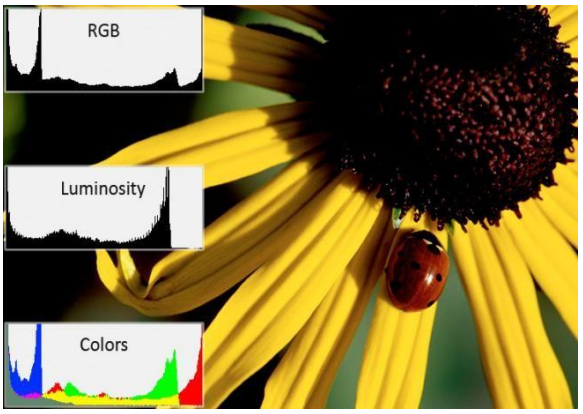


Fig 1.4.5.1 RGB histogram image

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multiresolution theory. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including sub band coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

1.4.6 Compression:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

1.4.7 Morphological processing:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all black pixels in a binary image is a complete morphological description of the image and shown in fig 1.4.7.1.

Fig 1.4.7.1 blur to de-blur image

In binary images, the sets in question are members of the 2-D integer space Z^2, where each element of a set is a 2-D vector whose coordinates are the (x,y) coordinates of a black(or white) pixel in the image. Gray-scale digital images can be represented as sets whose components are in Z^3. In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray-level value.

1.4.8 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.

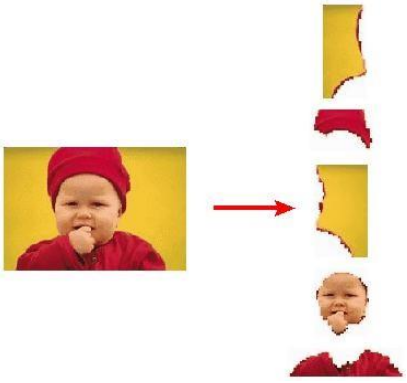


Fig 1.4.8.1 Image segmentation

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure shown in fig 1.4.8.1. In general, the more accurate the segmentation, the more likely recognition is to succeed.

1.4.9 Representation and description:

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections.

Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

1.5 Applications of Image Processing

Some of the major fields in which digital widely used image processing is are:

Medicine

Image segmentation and pattern recognition is used in digital mammography to identify tumors. Image registration and fusion play a crucial role in extracting information from medical image. Lossless compression used in telemedicine to transmit medical images from one place to another place effectively.

Forensics

Image processing is also used for security purpose. Different biometric used in identifications are face, fingerprint, iris, vein pattern etc. Edge enhancement, de-noising, skeletisation, etc. are used here.

Remote Sensing

The image processing used in the field of remote sensing include image enhancement, image merging and image classification techniques. Remote sensing is the use of the remote observations to make useful inference about a target.

Automobiles

'Night vision technology' is the latest invention in automobile sector. It helps to identify obstacles during night time and avoid accidents. Image enhancement, boundary detection and object recognition are the image processing techniques used here.

Communications

Now-a-days information can be easily transmitted through the internet because of the growth of multi-media technology. Effective image and video compression algorithms like JPEG, H.26X standards help to transmit the data effectively for live video conference.

1.6 COLOR IMAGE PROCESSING

1.6.2 Need of Color Transform

As per the medical research human eye will have different sensitivity to color and for brightness. Then it has come into the picture of having other color space other than the regular RGB color space.

1.6.3 Introduction

It is the technique used for application of transforms on the color images by transforming the color space. We have many color spaces available in the stream of Digital Image Processing such as RGB, HIS, YCbCr, CMY, CMYK, YIQ etc.

1.6.3.1 RGB

This is the basic color space for all other color spaces. All the color images we see in the daily life use RGB color space only. The image consists three basic planes termed as R-plane, G-plane, and B-plane shown in fig 1.6.3.1.1. Most of the information is present in G- plane after that in R-plane. Least information is present in B-plane. Hence, any changes that are needed can be made in this plane.

It is an additive color space. The complete specification of RGB color space requires a white point chromaticity and a gamma correction curve. The colors that are needed or obtained by mixing the R, G, and B components in the required levels, they are concatenated as shown in figure 1.6.3.1.2 and the important fact that has to be remembered is that, all the 3 color planes should always have same size in order to be concatenated each with others. Such image is the figure which has all three planes of same size and concatenated.

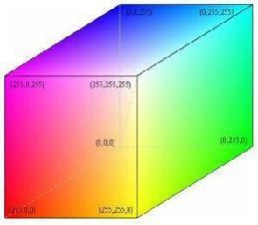


Fig 1.6.3.1.1 RGB color cube

Fig 1.6.3.1.2 RGB image

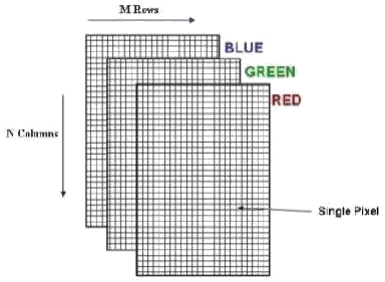


Fig 1.6.3.1.3 Image showing 3 different planes (R, G, B)

The color varies from Black to White diagonally. In this the basic colors that are used to obtain other colors are Red, Green, and Blue. RGB color image in which all the 3 planes (R, G, B) are concatenated with each other, in which same indexed values of different planes are having different values forming a single color image and shown in fig 1.6.3.1.3.

1.6.3.2 Lab Color Space

The Lab color space describes mathematically all perceivable colors in the three dimensions L for lightness and a and b for the color components green–red and blue–yellow. The terminology "Lab" originates from the Hunter 1948 color space. Nowadays "Lab" is frequently mis-used as abbreviation for CIEL*a*b* 1976 color space(also CIELAB); the asterisks/stars distinguish the CIEL version from Hunter's original version. The difference from the Hunter Lab coordinates is that the CIELAB coordinates are created by a cube root transformation of the CIE XYZ color data, while the Hunter Lab coordinates are the result of a square root transformation. Other, less common examples of color spaces with Lab representations make use of the CIE 1994 color difference and the CIE 2000 color difference.

The Lab color space exceeds the gamut's of the RGB and CMYK color models (for example, Pro Photo RGB includes about 90% of all perceivable colors). One of the most important attributes of the Lab model is device independence. This means that the colors are defined independent of their nature of creation or the device they are displayed on. The Lab color space is used when graphics for print have to be converted from RGB to CMYK, as the Lab gamut includes both the RGB and CMYK

gamut. Also it is used as an interchange format between different devices as for its device independency. The space itself is a three-dimensional real number space, which contains an infinite number of possible representations of colors. However, in practice, the space is usually mapped onto a three-dimensional integer space for device-independent digital representation, and for these reasons, the L*, a*, and b* values are usually absolute, with a pre-defined range. The lightness, L*, represents the darkest black at L* = 0, and the brightest white at L* = 100. The color channels, a* and b*, will represent true neutral gray values at a* = 0 and b* = 0. The red/green opponent colors are represented along the a* axis, with green at negative a* values and red at positive a* values. The yellow/blue opponent colors are represented along the b* axis, with blue at negative b* values and yellow at positive b* values. The scaling and limits of the a* and b* axes will depend on the specific implementation of Lab color, as described below, but they often run in the range of ±100 or −128 to
+127 (signed 8-bit integer).

Both the Hunter and the 1976 CIELAB color spaces were derived from the prior "master" space CIE 1931 XYZ color space shown in fig 1.6.3.2.1, which can predict which spectral power distributions will be perceived as the same color, but which is not particularly perceptually uniform. Strongly influenced by the Mansell color system, the intention of both "Lab" color spaces is to create a space that can be computed via simple formulas from the XYZ space but is more perceptually uniform than XYZ. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same perceptual distance. When storing colors in limited precision values, this can improve the reproduction of tones. Both Lab spaces are relative to the white point of the XYZ data they were converted from. Lab values do not define absolute colors unless the white point is also specified.

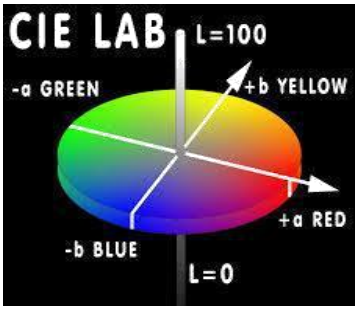


Fig 1.6.3.2.1 L*a*b* color image

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

The various approaches for detecting the disease in groundnut leaf using image processing technique is described in this section

Ramakrishnan.M [1] has proposed a calculation for leaf infection discovery. A standout amongst the most critical components adding to less yield is malady assault. The plant illness, for example, organisms, microbes and infections[1]. The leaf malady totally devastates the nature of the leaf. Shared groundnut illness is cercospora .It is unique of the nice of sickness in beginning period of groundnut leaf. The updated preparing design involves of four driving advances. At first a shading remodel establish planned for the info RGB picture is shaped, This RGB is transformed over into HSV on the grounds that RGB is for Color age and shading descriptor. The subsequent stage is plane partition. Following played out the shading highlights. At that point utilizing back proliferation calculation location of leaf infection is finished. The advantage is that, it comprises of four major steps. They are 1) Forming RGB image, 2) Converting RGB to HSV, 3) Plane seperation and 4) Back Propogation. The disadvantage is based on factors donating to less yield is disease attack. The leaf disease totally destroys the quality of the leaf.

Vijai Singh [2] present the image segmentation based on genetic algorithm. It identified in common as plant's disease detection. The advantage is based on the productivity of agriculture[2]. The disadvantage is that proper care need to be taken in disease detection or else it reasons sudden properties on plants and because of which the agricultural grade and amount or yield is affected.

Shweta R. [3] represents the common disease detection in plants[3].Common image processing algorithm that includes image segmentation, feature extraction and classification is used. The advantage is the disease is detected using the leaf images. The disadvantage is that it is very difficult to monitor the plant diseases manually.

R. P. L. DURGABAI [4] state that different machine learning technique are used to classify the disease[4] . The diseases like pest attack, diseases and climatic conditions are being identified. Advantage is the protection of the crops and the disadvantage is that the damage is created by other pest organisms to the agricultural crops.

R.Rajmohan [5] proposed a sensor based mobile app framework using neural network and SVM algorithm. This paper identified paddy yield and its condition in paddy crops[5]. The advantage is the accuracy in the agriculture business that provides valuable information. Sharada P.Mohanty et al implemented deep learning and neural network algorithms in smart phones. This identified 14 crop species and 26 diseases[6]. The advantage is the best accuracy and the disadvantage is the complication that occurs due to device failure and expensive.

David Heckmann et al[6] describe the partial least squares regression algorithm. It predicts certain parameters in the photosynthetic capacity. The advantage is that the properties of the reflectance spectar is systematically evaluated and provided a good performance.

S. Arivazhagan [7] implemented classification algorithm for detecting plant diseases. This detects the plant leaves diseases[10]. The advantage is the efficiency and it can also successfully identify and classify the diseases with high accuracy and robustness. The disadvantage is that the classification becomes wrong if the input is not provided clearly.

L.Yuan [8] performs the spectral analysis for winter wheat leaves in order to detect and differentiate both insects and diseases. Here, the potential of hyper spectral sensor systems was examined for differentiating three different stressors at leaf level. The measurement of reflectance spectra for leaves which were tainted with powdery mildew, yellow rust, and aphids was performed at the premature grain filling period. Hence, earlier to spectral analysis, author applied normalization on all three sets of samples in order to remove discrepancies in the spectral baseline among the all diverse cultivars.

S. Chabrier et. al [9] proposes optimization based image segmentation scheme with the use of genetic algorithms. The proposed technique was having an evaluation principle which was found to enumerate the quality of segmentation. The proposed scheme was able to integrate a local ground truth values when it was accessible for setting up the preferred level of precision for ultimate segmentation results. Along with this, the genetic algorithm was also used to decide the best combination of data taken out by the selected principle. This technique was found suitable for both gray level image and multi components image in not only supervised framework but also in unsupervised perspective.

J.G. Rosiles et. al [10] proposes multichannel texture segmentation based on the image pyramids created with the Bamberger directional filter bank.

The effect on segmentation performance was taken care of due to various factors such as the number of directional channels, the number of pyramid levels, redundancy and specifications of filter employed. The proposed technique also proved that the segmentation results which were obtained using the maximally decimated directional filter bank were adversary as those of the non-decimated directional filter bank.

Wang Jun and Wang Shitong [11] proposes a new segmentation scheme based on image-thresholding using Weighted Parzen-Window Estimation. Here, before the use of the weighted Parzen-window method, first hierarchical clustering technique was carried out to achieve the reference pixels and their respective weights which were used to approximate the consequent probabilities. The error created for the duration of generation of reference pixels was restricted by the upper bound error. Hence, with the use of the proposed criterion function, the best possible threshold was calculated. For speeding up the Parzen-window technique and to diminish the computational trouble, the author developed its superior version i.e., image thresholding using weighted Parzen-window estimation. Here, a training method was carried out to achieve a set of the reference pixels and weights for every value of gray level, which unexpectedly produced errors during the computation of optimal threshold.

Otsu, N [12] proposes threshold selection technique from gray-level histograms for segmentation. This was actually a nonparametric and unsupervised technique of automatic threshold selection for segmenting any image. An optimal threshold was obtained using discriminated principle, in order to make the most of the separation of the resultant sets in gray levels. The technique used here was quite straightforward, making use of simply the zeroth- and the first-order cumulative moments of calculated histograms for corresponding gray levels. This was proved to be quite simple to make extension of this technique for multi threshold issues too. This technique was found to be covering a variety of unsupervised decision methodologies.

Sezgin, M. and Sankur [13] collected a survey of different types of segmentation schemes and categorized them into six different groups in accordance with the data and information they were taking advantage of. However, the techniques were found to perform good only for the images with text document images and started degrading with noise and blur. Also, the images with patterned backgrounds such as checks, degradation in images due to non-uniform illumination and shadowing or color shades, or other kinds of mixed up or messed up in images were not found to perform up to mark. In addition to this, the growing number of color-images was also found to be the key challenging aspect for segmentation.

Weizheng, S [14] proposes grading technique for detecting spot diseases in leaf. The success or failure of the research was highly dependable on precise segmentation of lesion area. There were few features which were found to be controlling parameters for experiments such as variation in luminance level, presence of leaf vein, instability of lesion area characteristics etc. Also, during various phases of the disease, lesion area was found to reflect a variety of symptoms due to influence of air, water, light, nutrition etc. and hence resulting in obstacles during segmentation. The Author transformed the image from RGB to HSI color space and performed Otsu thresholding. Along with this in order to segment lesion area precisely, the author performed Sobel edge transformation for extraction of lesion edges, area filling and morphological operations on the H channel of HSI color space. D Lorente [15] worked on selection of optimal wavelength features for detection of decay detection of decay in citrus fruit using the ROC curve but classification accuracy was found to be quite minimal.

Z Zhong et. al [16] proposes an adaptive background modeling scheme for foreground segmentation. This was performed using combined usage of the pixel-based adaptive segmentation technique along with updating scheme of the background. This was conducted not only on pixel level but also on object level. The motivation behind this work was to overcome the barriers from the present pixel-based adaptive segmentation techniques which were only able to update the background only at the pixel level not on the object level if any changes occur in objects. These barriers were resulting issues in detection of foreground, object update speeds etc. To overcome these problems, author utilized a counter which was able to place the foreground pixels into two different categories i.e illumination and object. The proposed method was found to extract an accurate foreground object while having the control over update time of pixels which were related to the object or an illumination region respectively.

P. Rani et. al [17] proposes retinal vessel segmentation scheme under pathological conditions using supervised machine learning. The author also used matched filter scheme along with first-order derivative of the Gaussian to enhance the segmentation of the retinal blood vessels which were occurring as a result of cross sectional match of the respective blood vessels to the their corresponding Gaussian profile. However, the limitation of the proposed technique was found when the non-vessel patterns were attached to the vessel patterns upon matched filter scheme along with first-order derivative of the Gaussian segmentation technique; they could not be masked out.

P. Li, B.Hu [18] segments very high resolution images using supervised region merging scheme based on binary classification and active learning approach. This technique was found to perform the job of making a decision if two contiguous segments should have been merged as a binary classification issue. The support vector machine (SVM) technique was applied as the binary classifier during this research.

S. Li, et. al [19] implements segmentation scheme for Spine MRI images. Here, author proposed a completely automatic, unsupervised segmentation scheme for inter vertebral disc with the extensive use of enhanced marker controlled watershed transformation in order to combine both intention and prior shape data. This technique was able to produce closed contour, combining gray levels and priori data for image in order to enhance the robustness

A. Gupta et. al [20] proposes adaptive thresholding scheme in order to segment the skin lesions with the use of Statistical Parameters. Different statistical features such as mean and standard deviation were used to preprocess and hence segment the lesion areas absolutely in an automatic fashion. Average filtering technique for exclusion of hair and skin scales and mathematical morphological techniques for discarding the false positives were also applied to enhance the accuracy of the proposed segmentation scheme.

Y. Zhuang et. al [21] proposes feature-aligned segmentation scheme using correlationclustering technique. The author proposed a methodology to segment a lattice into patches whose boundaries were associated with major ridge and valley lines of the contour. The main motivation was the formulation of this issue as correlation clustering (CC), a graph partitioning issue which initiates from the data mining society. This key formulation provided two inimitable advantages to this research over available segmentation schemes. First, since CC was absolutely parametric, this technique was found to have control over few parameters. Second, as CC was only governed by edge weights in graph, but this technique was able to provide users straight and local control over the segmentation result. The author constructed weighted graph on which CC was defined, and developed a unique approach for swiftly computing CC for the corresponding graph, along with an interactive toolset for editing the segmentation.

Problem Statement:

Groundnut will be harvested twice in a year. Most of groundnut farmer faces many problems to harvest the groundnut because they had been attack by snail, worm and fungi. Furthermore, when the groundnut had been infected or attacked, the others areas had been exposed to be infected. Thus, it will decrease yield of the crop and lead to significance losses to farmer. Currently, the groundnut farmer determines the type of disease manually. The errors might occur in order to determine the type of diseases. groundnut farmer also have to spend a lot of time to detect the type of disease. It also takes a time as the groundnut farmers manually check the disease since the groundnut field is in wide area.

CHAPTER -3

GROUNDNUT OVERVIEW

GROUNDNUT OVERVIEW

In this section, firstly presents a definition of groundnut. After that, this subsection briefly discusses on type of groundnut disease, symptoms and management of groundnut disease.

3.1 Definition of groundnut

The peanut, also known as the groundnut, goober (US), or monkey nut (UK), and taxonomically classified as *Arachis hypogaea*, is a legume crop grown mainly for its edible seeds. It is widely grown in the tropics and subtropics, being important to both small and large commercial producers. It is classified as both a grain legume and, due to its high oil content, an oil crop. World annual production of shelled peanuts was 44 million tonnes in 2016, led by China with 38% of the world total. Atypically among legume crop plants, peanut pods develop underground (geocarpy) rather than above ground. With this characteristic in mind, the botanist Carl Linnaeus named the species *hypogaea*, which means "under the earth". This cereal crop that is widely cultivated in warm climates for its seeds and by-products. Groundnut is one of the most utilized food plants and widely grown originated in ASIA. Groundnut is an important crop worldwide and over half of the world population relies on it for oil and other purposes. Many people in the world including India take Groundnut as edible food.

3.1.1 Groundnut Diseases, Symptoms and Management

There are many factors that make groundnut production become slow and less productive. One of the main factors is groundnut disease. The table below will show you type of groundnut disease, the symptom of groundnut disease and the management of groundnut disease. This researches focus on four types of diseases, which are bacterial sheath blight, early leaf spot, bacterial rot and late leaf spot.

A. Early leaf spot:

Cercospora leaf spot disease is a disease caused by fungi on leaves of groundnut. It is a major problem, although traditional farmers in Sierra Leone consider spots on groundnut as signs of maturity. In most cases, farmers find it difficult to clearly differentiate between symptoms of early and late leaf spot of groundnut. Brown lesions (spots), usually surrounded by a yellow color on the upper side of leaves, are the most common symptom of early leaf spot. Dark brown lesions (spots), usually on the underside of affected leaves, are the most common symptom of late leaf spot shown in fig 3.1.1.1. It is important to determine if late leaf spot is present since it can be difficult to control.

Fig 3.1.1.1 early leaf spot

B. Late leaf spot:

Late leaf spot (LLS) and rust are two major foliar diseases of peanut that often occur together leading to 50–70% yield loss in the crop. The symptoms of LLS are circular and darker spots appear on the lower surface of the leaves and also forms in stems and pegs resulting in severe yield loss to the groundnut growers. LLS also have an adverse influence on seed quality as well as on quality of haulms shown in fig 3.1.1.2. The regular incidence of the above disease under late sown conditions warrants the development of resistant cultivars in groundnut. Though there are many chemical control methods available, development of disease resistant varieties is the best way to control the disease to improve production quality and reduce the adverse effects of chemicals on our ecosystem.

Fig 3.1.1.2 late leaf spot

C. Rust

This disease occurs in Central and South America, West Indies, Venezuela, China, erstwhile U.S.S.R., U.S.A. and India. In India it was first observed in 1971 in Punjab, Tamil Nadu, West Bengal mand Andhra Pradesh. This has become one of the serious diseases of groundnut.

Symtoms appers as, Orange colored pustules (uredinia) on the lower leaflet surface. Rupture as masses of reddish brown urediniospores. Pustules appear first on the lower surface of leaflet. Pustules may also appear on the upper surface of the leaflet. Severely infected leaves turn necrotic and desiccate but are attached to the plant shown in fig 3.1.1.3.

Fig 3.1.1.3 Rust

D. Bud Necrosis

Terminal buds of plants are affected when temperatures are relatively high. Leaflets produced on auxiliary shoots show a wide range of symptoms including reduced size, distortion of the lamina, mosaic, and general chlorosis. Early infection results in stunting of plants due to multiplication of affected terminal buds. Any seeds produced by early-infected plants are small, shriveled, and with spots. First appear on young leaves as necrotic lesions and vertical necrosis shown in fig 3.1.1.4. The necrosis later spreads to the petiole and stem. Necrotic lesions on the stem later spread upwards killing the bud. In some cultivators, pods harvested from the PSDN infected plants show necrotic lesions.

Fig.3.1.1.4 Bud Necrosis

3.2 Method

The methodology for diagnosing groundnut diseases can be simplified. This process involves several tasks, such as image acquisition and collection, image segmentation and pre-processing, shape feature extraction and color feature extraction, and groundnut diseases classification based on lesion type, boundary color, spot color, and broken groundnut leaf color.

3.2.1 Image Acquisition

The RGB color images of groundnut leaf are captured using a Canon Power Shot G2 digital camera, with pixel resolution 768x1024. The digitized images are about 225 KB size each. Those images are cropped into a smaller image with dimension of 109 x 310 pixels. There have collected about 94 data samples. It consists of three types of groundnut diseases as shown in figures. Images are stored in BMP format. The prototype uses Matlab image processing library

3.2.2 Color Transformation

Color can be described by its red (R), green (G) and blue (B) coordinates (the well-known RGB system), or by some its linear transformation as XYZ, CMY, YUV, IQ, among others. The CIE adopted systems CIELAB and CIELUV, in which, to a good approximation, equal changes in the coordinates result in equal changes in perception of the color. Nevertheless, sometimes it is useful to describe the colors in an image by some type of cylindrical-like coordinate system, it means by its hue, saturation and some value representing brightness. If the RGB coordinates are in the interval from 0 to 1, each color can be represented by the point in the cube in the RGB space. Let us imagine the attitude of the cube, where the body diagonal linking ”black” vertex and ”white” vertex is vertical. Then the height of each point in the cube corresponds to the brightness of the color, the angle or azimuth corresponds to the hue and the relative distance from the vertical diagonal corresponds to the saturation of the color.

3.2.3 K-Means Clustering Method

K-means clustering is used to partition the leaf image into four clusters in which one or more clusters contain the disease in case when the leaf is infected by more than one disease. The k-means clustering algorithms tries to classify objects (pixels in our case) based on a set of features into K number of classes. The classification is done by minimizing the sum of squares of distances between the objects and the corresponding cluster or class centroid. On this experiment, the K-means clustering is set to use squared Euclidean distances.

Image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analysis. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture

3.2.4 Feature Extraction

The image analysis focused on the shape feature extraction and color based segmentation. The method followed for extracting the feature set is called the Gray level Co-occurrence Method or GLCM method in short. It is a method, in which both the color and texture of an image are taken into account, to arrive at unique features, which represent that image.

3.2.5 Image

The production rules have been developed through serial interviews with agricultural expert based on above characteristics, such boundary color, spot color, and broken groundnut leaf color, groundnut diseases is recognized using K-Means Clustering method.

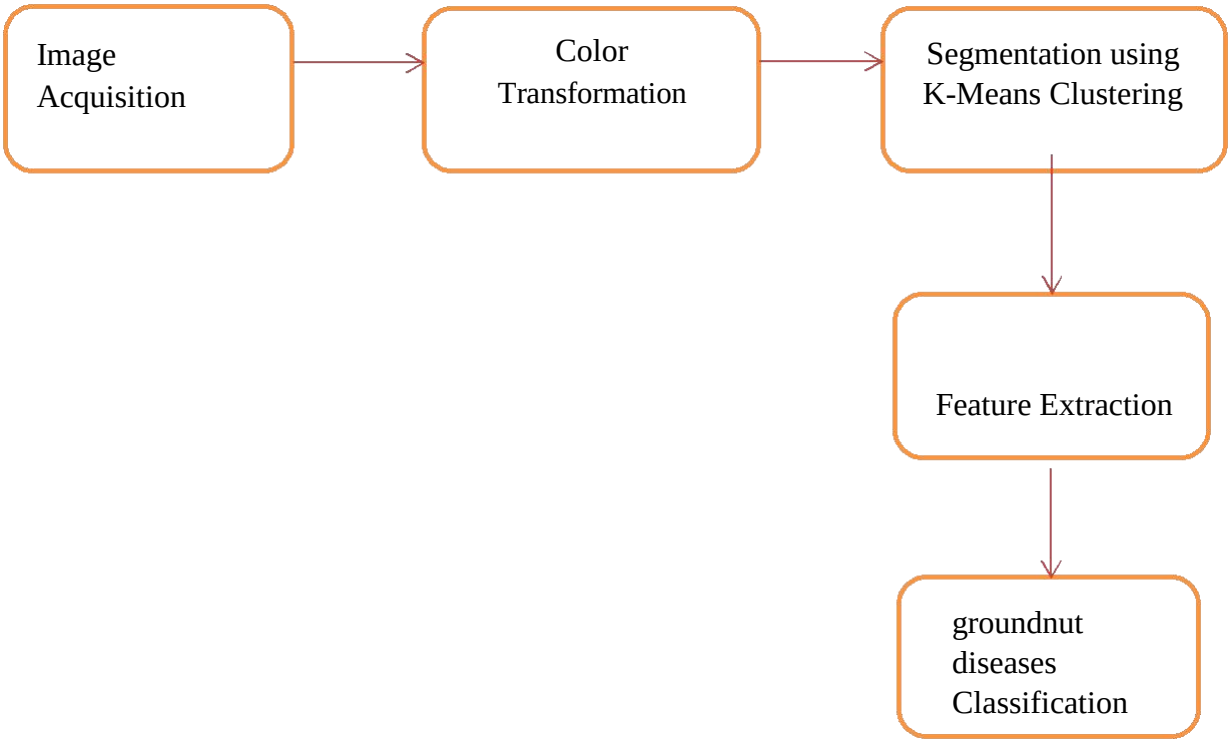
CHAPTER-4

GROUNDNUT DISEASE CLASSIFIACTION

4.1 GROUNDNUT DISEASE CLASSIFIACTION

The disease is classified in groundnut at early stages using K-Means clustering methodology. In groundnut diseases cannot be detected at early stages. Hence it creates a great loss for the farmers. In order to save the damage of the yield using these methodology diseases can be detected at early stages. Many diseases occur in groundnut, but this method detects four types of disease they are early leaf spot, late leaf spot, Rust, Bud necrosis.

4.1 Process of groundnut Disease Detection

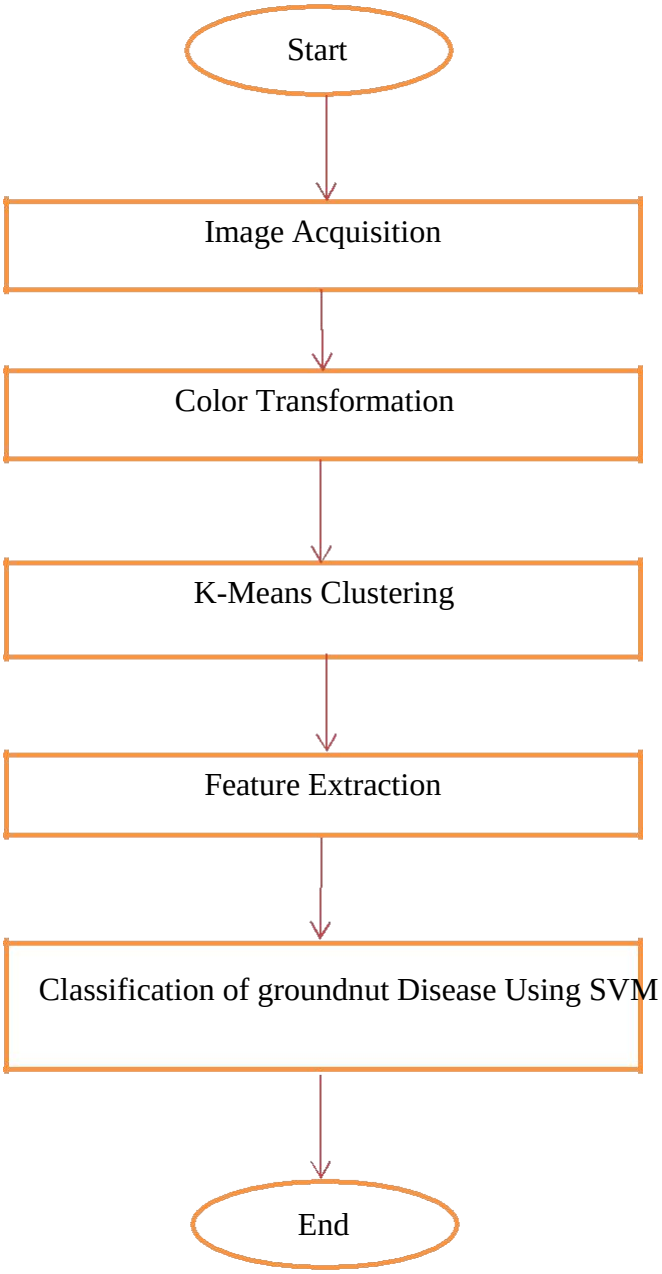


4.1.1 Block diagram of groundnut Disease Detection

4.2 Methodology for disease classification in groundnut

The captured images are taken as input. Then the image is converted from RGB into LAB image and then K-Means clustering is used followed by feature extraction. In feature extraction disease can be detected based on the shape. In the next step disease is classified.

.



4.2.1 Flow Chart of Methodology

A. Image Acquisition

Image Acquisition shows the fundamental steps in image processing. In this the first step is image acquisition. The process of capturing the real world image and storing it into a computer is called image acquisition. Conventional silver-based photographs in the form of negatives, transparencies or prints can be scanned using a variety of scanning devices. Digital cameras which capture images directly in digital form are more popular now-a-days. They use charge coupled arrays as the image sensors that converts light into electrical signal.

B. Color Transformation

A Lab color space is a color opponent space with dimension L for lightness and a and b for the color opponent dimensions, based on nonlinearly compressed CIE XYZ color space coordinates. A "Lab" color space is to create a space which can be computed via simple formulas from the XYZ space, but is more perceptually uniform than XYZ. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same visual importance. When storing colors in limited precision values, this can improve the reproduction of tones. Both Lab spaces are relative to the white point of the XYZ data they were converted from. Lab values do not define absolute colors unless the white point is also specified. Your goal is to identify different colors in image by analyzing the L*a*b* color space. The image was acquired using the Image Acquisition Toolbox.

C. K-Means clustering

Clustering is a method to divide a set of data into a specific number of groups. It's one of the popular method is k-means clustering. In k-means clustering, it partitions a collection of data into a k number group of data11, 12. It classifies a given set of data into k number of disjoint cluster. K-means algorithm consists of two separate phases. In the first phase it calculates the k centroid and in the second phase it takes each point to the cluster which has nearest centroid from the respective data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done it recalculate the new centroid of each cluster and based on that

Centroid a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster which have minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. So K-means is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, overall clusters. Let us consider an image with resolution of x, y and the image has to be cluster into k number of cluster. Let p(x, y) be an input pixels to be cluster and k be the cluster centers.

The algorithm for k-means is in the following:

1. Initialize number of cluster k and center.

2. For each pixel of an image, calculate the Euclidean distance d, between the center and each pixel of an image using the relation given below.

3. Assign all the pixels to the nearest center based on distance d.

4. After all pixels have been assigned, recalculate new position of the center using the relation given below.

5. Repeat the process until it satisfies the tolerance or error value.

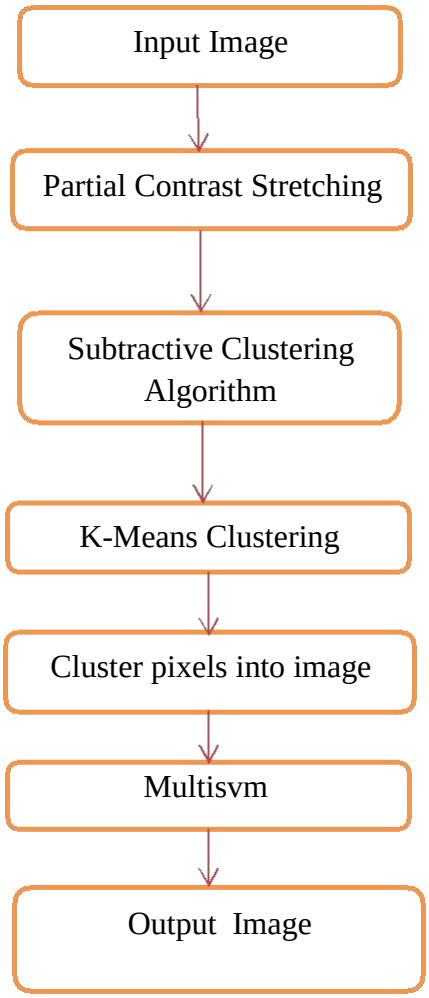
6. Reshape the cluster pixels into image.

Although k-means has the great advantage of being easy to implement, it has some drawbacks. The quality of the final clustering results is depends on the arbitrary selection of initial centroid. So if the initial centroid is randomly chosen, it will get different result for different initial centers. So the initial center will be carefully chosen so that we get our desire segmentation shown in fig 4.2.2.

Mostly the medical images which are used for segmentation have low contrast. So contrast stretching is used to improve the quality of the image. After improving the quality of image, subtractive clustering algorithm is used to generate the centers, based on the potential value of the image. Number of center is generated based on number of cluster k. This center is used as initial center in k-means algorithm. Using the k-means algorithm, the image is segmented. After the

Segmentation of image, the image can still contain some unwanted region or noise. These noises are removed by using the median filter.

1. Load the image to be segmented.
2. Apply partial contrast stretching. Initialize number of cluster k.
3. Calculate the potential for every pixel value of the image
4. Find maximum potential in step 3 and set that point be first center cluster and its corresponding potential as maximum potential.
5. Update the potential value of other remaining pixels based on the first cluster center.



4.2.2 Flow chart of image clustering

D. Segmentation

The image segmentation is defined as an optimal segmentation obtained in a pure bottom-up fashion that provides the information necessary to initialize and constrain high-level segmentation methods. Although the details of primary segmentation methods will depend on the application domain, we require that they do not depend on a priori knowledge about the objects present in a particular scene or image specific parameter adjustments. These claims become realistic because we do not seek for a perfect segmentation result but rather for the best possible support for more intelligent methods to be applied afterwards. Unfortunately up to now there is no theory which defines the quality of a segmentation. Therefore we have to rely on some heuristic constraints which the primary segmentation should meet: The segmentation should provide regions that are homogenous with respect to one or more properties, i.e. the variation of measurements within the regions should be considerably less than the variation at borders. The position of the borders should coincide with local maxima, ridges and saddle points of the local gradient of the measurements. Areas that perceptually form only one region should not be splitted into several parts. In particular this applies to smooth shading and texture. Small details, if clearly distinguished by their shape or contrast, should not be merged with their neighbouring regions.

E. Feature Extraction

After the images are pre-processed, the features of the groundnut disease need to be extracted to be fed to the system for recognition which is the shape of the disease. This process is also done in MATLAB which uses morphological operations. The feature extraction of the image is done from the continuous steps of pre-processing the images. First step is the process of converting the images from RGB color profile to Grayscale color profile. Once done, the next step is to detect the edges of the defect. Canny Edge Detection was used in this project. A series of experiments were conducted that lead to decision of using Canny Edge Detection in this project. The last step in this stage is smoothing the image. This is done to enhance the image's features. In this step, a combination of image dilation and erosion was used. In Feature extraction some of the parameters are calculated in order to classify groundnut leaf diseases.

(i) Skewness

In terms of digital image processing darker and glossier surfaces tend to be more precisely skewed than lighter and made surfaces. Hence we can use skewness in making judgements about image surfaces.

(ii) Contrast

Contrast is the difference in luminance or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in color and brightness of the objects and other objects within the same field of view.

(iii) Correlation

Correlation is a basic operation that we will perform to extract information from images. They are in some sense the simplest operations that we can perform on an image, but they are extremely useful.

(iv) Energy

Energy is used to describe a measure of information than formulating an operation under a probability framework. Energy is a fairly loose term used to describe any user defined function (in the image domain). The motivation for using the term 'Energy' is that typical object detection or segmentation tasks are posed as a Energy minimization problem

(v) Homogeneity

Homogeneity is a concept often used to represent the uniformity in a material or substance.

(vi) Standard Deviation

Standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of values. In statistics, the standard deviation (SD, also represented by the Greek letter sigma σ or the Latin letter s) is a measure that is used

to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values.

The standard deviation of a random variable, statistical population, data set or probability distribution is the square root of its variance. It is algebraically simpler, though in practice less robust, than the average absolute deviation. A useful property of the standard deviation is that, unlike the variance, it is expressed in the same units as the data. There are also other measures of deviation from the norm, including average absolute deviation, which provide different mathematical properties from standard deviation.

(vii) Entropy

Entropy is a quantity which is used to describe the amount of information which must be coded for a compression algorithm. In statistical mechanics entropy is related to the number of microscopic configurations Ω that a thermodynamic system can have when in a state as specified by some macroscopic variables. Specifically, assuming for simplicity that each of the microscopic configurations is equally probable, the entropy of the system is the natural logarithm of that number of configurations, multiplied by the Boltzmann constant k_B.

(viii) RMS

Root mean square value is a frequently used measure of the predicted by a model or an estimator and the values actually observed. In statistics and its applications, the root mean square (abbreviated RMS or rms) is defined as the square root of mean square (the arithmetic mean of the squares of a set of numbers). The RMS is also known as the quadratic mean and is a particular case of the generalized mean with exponent 2. RMS can also be defined for a continuously varying function in terms of an integral of the squares of the instantaneous values during a cycle.

(ix)Variance(σ^2)

Variance is the expectation of the squared deviation of a random variable from its mean. In probability theory and statistics, variance is the expectation of the squared deviation of a random variable from its mean. Informally, it measures how far a set of (random) numbers are spread out from their average value. Variance has a central role in statistics, where some ideas that use it include descriptive statistics, statistical inference, hypothesis testing, goodness of fit, and Monte Carlo sampling. Variance is an important tool in the sciences, where statistical analysis of data is common. The variance is the square of the standard deviation, the second central moment of a distribution, and the covariance of the random variable with itself.

(x) Smoothness

Smoothness in image processing is used to remove the noise. Smoothing is often used to reduce noise within an image or to produce a less pixelated image. Most smoothing methods are based on low pass filters. Smoothing is also usually based on a single value representing the image, such as the average value of the image or the middle (median) value.

(xi) Kurtosis

In image processing kurtosis values are interpreted in combination with noise and resolution measurement.

In probability theory and statistics, kurtosis (from, *kyrtos* or *kurtos*, meaning "curved, arching") is a measure of the "tailedness" of the probability distribution of a real-valued random variable. In a similar way to the concept of skewness, *kurtosis* is a descriptor of the shape of a probability distribution and, just as for skewness, there are different ways of quantifying it for a theoretical distribution and corresponding ways of estimating it from a sample from a population. Depending on the particular measure of kurtosis that is used, there are various interpretations of kurtosis, and of how particular measures should be interpreted.

The standard measure of kurtosis, originating with Karl Pearson, is based on a scaled version of the fourth moment of the data or population. This number is related to the tails of the distribution, not its peak; hence, the sometimes-seen characterization as "peakedness" is mistaken. For this measure, higher kurtosis is the result of infrequent extreme deviations (or outliers), as opposed to frequent modestly sized deviations.

The kurtosis of any univariate normal distribution is 3. It is common to compare the kurtosis of a distribution to this value. Distributions with kurtosis less than 3 are said to be platykurtic, although this does not imply the distribution is "flat-topped" as sometimes reported. Rather, it means the distribution produces fewer and less extreme outliers than does the normal distribution. An example of a platykurtic distribution is the uniform distribution, which does not produce outliers. Distributions with kurtosis greater than 3 are said to be *leptokurtic*. An example of a leptokurtic distribution is the Laplace distribution, which has tails that asymptotically approach zero more slowly than a Gaussian, and therefore produces more outliers than the normal distribution. It is also common practice to use an adjusted version of Pearson's kurtosis, the excess kurtosis, which is the kurtosis minus 3, to provide the comparison to the normal distribution. Some authors use "kurtosis" by itself to refer to the excess kurtosis. For the reason of clarity and generality, however, this article follows the non-excess convention and explicitly indicates where excess kurtosis is meant.

(xii) Mean

Mean is calculating the sum of the valu1es by total number of values. In probability and statistics mean and expected value are used synonymously to refer to one measure of the central tendency either of a probability distribution or of the random variable characterized by that distribution.[1] In the case of a discrete probability distribution of a random variable *X*, the mean is equal to the sum over every possible value weighted by the probability of that value; that is, it is computed by taking the product of each possible value *x* of *X* and its probability P(*x*), and by adding all the values we get mean value.

F. Support Vector Machine

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations.

For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation.

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "kernel trick". Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors.

Algorithms capable of operating with kernels include the kernel perceptron, support vector machines (SVM), Gaussian processes, principal components analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters and many others. Any linear model can be turned into a non-linear model by applying the kernel trick to the model: replacing its features (predictors) by a kernel function.

Most kernel algorithms are based on convex optimization or eigen problems and are statistically well-founded. Typically, their statistical properties are analyzed using statistical learning theory (for example, using Rademacher complexity).

CHAPTER-5

RESULTS

5. RESULTS

The groundnut production can be hampered as effect of some nutritional deficiency, genetically disorder, climatic conditions etc. The major problem is disease causing by macrobes and microbes. Due, to groundnut leaf diseases there is a loss in yield of groundnut. Now a days there are a lots of groundnut leaf diseases, but in this work we have taken four diseases as our experimental model. The diseases are Early leaf spot, Late leaf spot, Rust , Bud Necrosis.

Image Acquisition is to capture the image of groundnut leaves. Color Transformation is to convert SRGB image to L*a*b* image. In K-Means Clustering is segmentation is done using K-Means Clustering. Feature extraction is used for the features of the groundnut leaf diseases need to be extracted to fed to system for recognition of shape of disease. groundnut Diseases Classification is done on the basis of feature extraction.

5.1 Image Acquisition

Image Acquisition shows the fundamental steps in image processing. In this the first step is image acquisition. The process of capturing the real world image and storing it into a computer is called image acquisition. Conventional silver-based photographs in the form of negatives, transparencies or prints can be scanned using a variety of scanning devices. Digital cameras which capture images directly in digital form are more popular now-a-days shown in fig 5.1.1.

(a)

(b)

(c)

(d)

Fig 5.1.1 Groundnut diseased images

5.2 Color Transformation

A Lab color space is a color opponent space with dimension L for lightness and a and b for the color opponent dimensions, based on nonlinearly compressed CIE XYZ color space coordinates. A “Lab” color space is to create a space which can be computed via simple formulas from the XYZ space, but is more perceptually uniform than XYZ. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same visual importance. When storing colors in limited precision values, this can improve the reproduction of tones. Both Lab spaces are relative to the white point of the XYZ data they were converted from. Lab values do not define absolute colors unless the white point is also specified. Your goal is to identify different colors in image by analyzing the L*a*b* color space.

Both the Hunter and the 1976 CIELAB color spaces were derived from the prior "master" space CIE 1931 XYZ color space, which can predict which spectral power distributions will be perceived as the same color, but which is not particularly perceptually uniform. Strongly influenced by the Mansell color system, the intention of both "Lab" color spaces is to create a space that can be computed via simple formulas from the XYZ space but is more perceptually uniform than XYZ. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same perceptual distance. When storing colors in limited precision values, this can improve the reproduction of tones. Both

Lab spaces are relative to the white point of the XYZ data they were converted from. Lab values do not define absolute colors unless the white point is also specified. Often, in practice, the white point is assumed to follow a standard and is not explicitly stated (e.g., for "absolute colorimetric" rendering intent, the International Color Consortium L*a*b* values are relative to CIE standard illuminate, while they are relative to the unprinted substrate for other rendering intents).

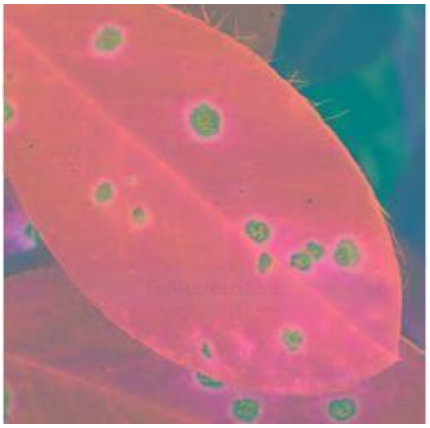


Fig 5.2.1 L*a*b* Color Space

5.3 K-Means Clustering

Clustering is a method to divide a set of data into a specific number of groups. It's one of the popular method is k-means clustering. In k-means clustering, it partitions a collection of data into a k number group of data11, 12. It classifies a given set of data into k number of disjoint cluster. K-means algorithm consists of two separate phases. In the first phase it calculates the k centroid and in the second phase it takes each point to the cluster which has nearest centroid from the respective data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done it recalculate the new centroid of each cluster and based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster which have minimum Euclidean distance. Each cluster in the

Partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. So K-means is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, overall clusters shown in fig 5.3.1. Let us consider an image with resolution of x × y and the image hast obecluster into k number of cluster. Let p(x, y) be an input pixels to be cluster and ck be the cluster centers.

Step 1: Given n objects, initialize k cluster centers.

Step 2: Assign each object to its closest cluster center.

Step 3: Update the center for each cluster.

Step 4: Repeat 2 and 3 until no change in each cluster center.

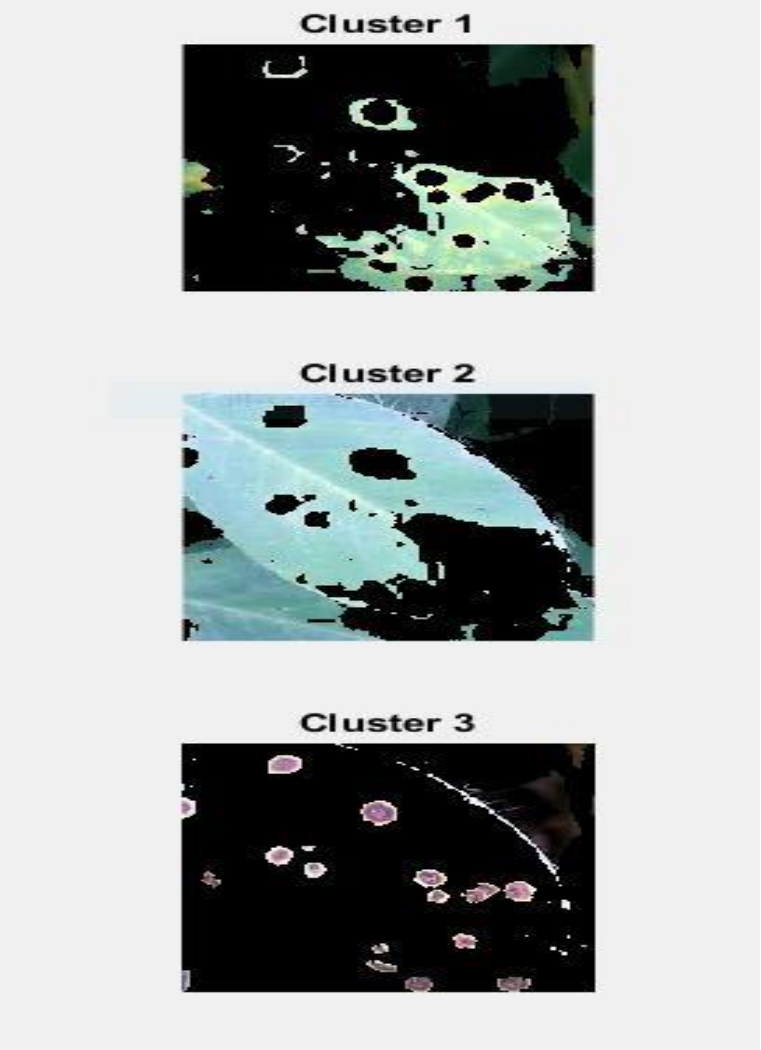


Fig 5.3.1 K-Means clustering

5.4 Feature extraction

After the images are pre-processed, the features of the groundnut disease need to be extracted to be fed to the system for recognition which is the shape of the disease. This process is also done in MATLAB which uses morphological operations. The feature extraction of the image is done from the continuous steps of pre-processing the images. First step is the process of converting the images from RGB color profile to Grayscale color profile. Once done, the next step is to detect the edges of the defect. Canny Edge Detection was used in this project. A series of experiments were conducted that lead to decision of using Canny Edge Detection in this project. The last step in this stage is smoothing the image. This is done to enhance the image's features shown in fig 5.4.1. In this step, a combination of image dilation and erosion was used

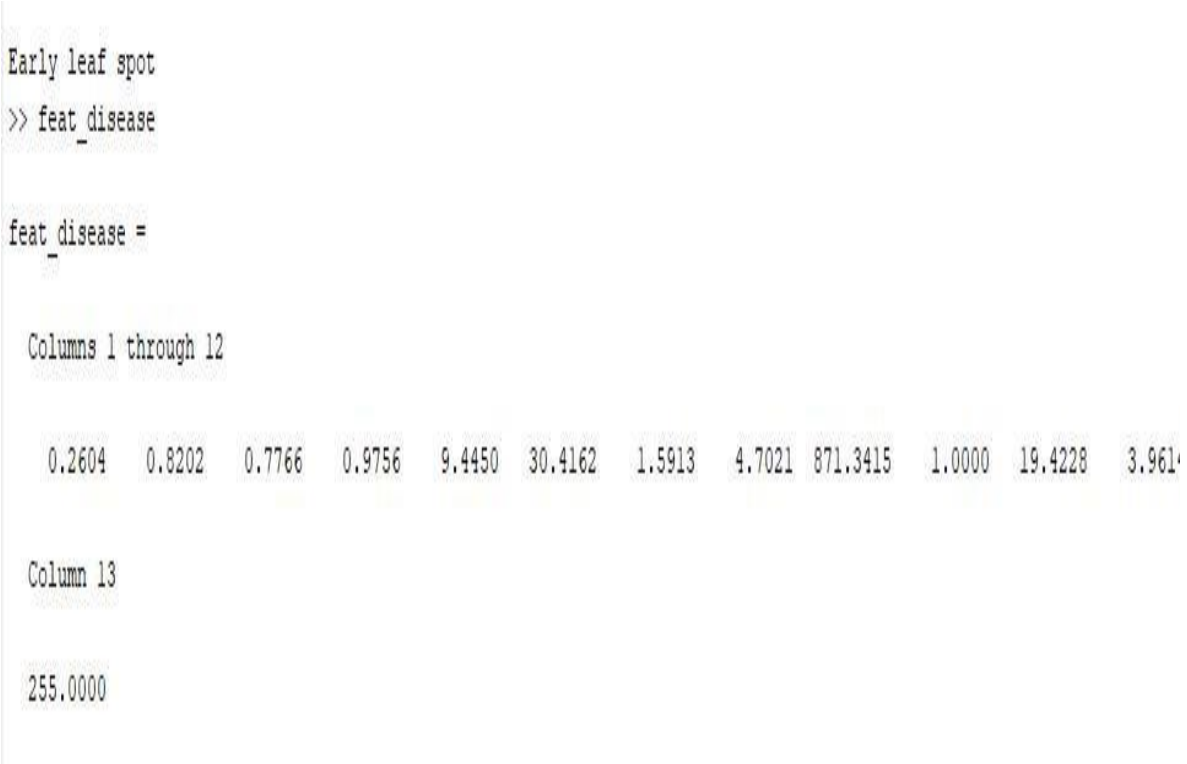


Fig 5.4.1 Feature Extraction

5.5 Groundnut Diseases Classification

The groundnut diseases are classified based on the SVM train algorithm. In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations. For example clusters, rankings, principal components, correlations, classifications in datasets. For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation.

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "kernel trick". Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors.

Algorithms capable of operating with kernels include the kernel perceptron, support vector machines (SVM), Gaussian processes, principal components
Analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters and many others. Any linear model can be turned into a non-linear model by applying the kernel trick to the model: replacing its features (predictors) by a kernel function.

K means clustering is used for segmentation and classification of groundnut diseases. The main usage of K-Means clustering is accurate and fast detection of groundnut leaf disease diseases like early leaf spot, late leaf spot, rust, Bud necrosis takes place. The proposed approach is image processing based. Due to this work, the groundnut leaf diseases can be identified at initial stage.

```
Early leaf spot
>> feat_disease

feat_disease =

  Columns 1 through 12

    0.2604    0.8202    0.7766    0.9756    9.4450   30.4162    1.5913    4.7021  871.3415    1.0000   19.4228    3.961

  Column 13

  255.0000
```

Fig.5.5.1 Output for Early Leaf Spot disease

5.1 Table of Feature extraction values of Early Leaf Spot

Contrast	0.260
Correlation	0.820
Energy	0.776
Homogeneity	0.975
Mean	9.445
Standard Deviation	30.41
Entropy	1.591
RMS	4.702
Variance	8.713
Smoothness	0.99
Kurtosis	19.422
Skewness	3.961
IDM	255

```
Late leaf spot
>> feat_disease

feat_disease =

   1.0e+03 *

  Columns 1 through 12

    0.0002    0.0009    0.0008    0.0010    0.0117    0.0427    0.0011    0.0035    1.5547    0.0010    0.0160    0.0038

  Column 13

    0.2550
```

Fig 5.5.2 Output for Late leaf spot

5.2 Table of Feature extraction values of Late leaf spot:

Feature Name	Value
Contrast	0.0004
Correlation	00009
Energy	0.0008
Homogeneity	0.0010
Mean	0.0117
Standard Deviation	0.0011
Entropy	0.0035
RMS	1.5547
Variance	0.0010
Smoothness	1.000
Kurtosis	0.0160
Skewness	0.0038
IDM	0.255

```
Rust
>> feat_disease

feat_disease =

   1.0e+03 *

  Columns 1 through 12

    0.0010    0.0008    0.0005    0.0009    0.0272    0.0533    0.0028    0.0072    2.1505    0.0010    0.0058    0.0019

  Column 13

    0.2550
```

Fig 5.5.3 Output for Rust

5.3 Table of Feature extraction values of Rust

Feature Name	Value
Contrast	0.0010
Correlation	0.0008
Energy	0.005
Homogeneity	0.0009
Mean	0.0272
Standard Deviation	0.0533
Entropy	0.0028
RMS	0.0072
Variance	2.1505
Smoothness	0.0010
Kurtosis	0.0058
Skewness	0.0019
IDM	0.2550

```
>> feat_disease

feat_disease =

  Columns 1 through 12

    0.0569    0.9317    0.4252    0.9728   23.1287   27.5905    3.8091   10.5808  649.1893    1.0000    1.7596    0.6055

  Column 13

  255.0000

fx >>
```

Fig 5.5.4 Output for Sheath Rot

5.4 Table of Feature extraction values of Sheath Rot

Feature Name	Value
Contrast	0.0569
Correlation	0.9317
Energy	0.4252
Homogeneity	0.9728
Mean	23.1287
Standard Deviation	27.5905
Entropy	3.8091
RMS	10.5808
Variance	649.1893
Smoothness	1.000
Kurtosis	1.7596
Skewness	0.6055
IDM	0.2550

CONCLUSION

In this project, K means clustering technique is used for segmentation and classification of groundnut diseases. This method gives accurate and fast detection of groundnut leaf diseases. It classifies the diseases such as blast, late leaf spot, sheath blight and bacterial rot. The proposed approach is image processing based. A set of groundnut leaf images is taken as input. The processing is done by using K- Means clustering and it divides the image into different segments which helps in accurate analysis of disease. The diseases are classified based on the support vector machine.

Future Scope

The future scope of groundnut leaf detection using segmentation through k-means clustering, the disease can be detected in the early stages of groundnut leaf. It helps the farmer to take necessary precautions and to increase the yield of the crop.

References

[1] S. J. G. A. Barbedo, “A new automatic method for disease symptom segmentation in digital photographs of plant leaves,” European Journal of Plant Pathology, vol. 147, no. 2, pp. 349–364, 2016.

[2] J. G. A. Barbedo, “A novel algorithm for semi-automatic segmentation of plant leaf disease symptoms using digital image processing,” Tropical Plant Pathology, vol. 41, no. 4, pp. 210– 224, 2016.

[3] J. G. A. Barbedo, L. V. Koenigkan, and T. T. Santos, “Identifying multiple plant diseases using digital image processing,” Biosystems Engineering, vol. 147, pp. 104–116, 2016.

[4] J. Pang, Z.-Y. Bai, J.-C. Lai, and S.-K. Li, “Automatic segmentation of crop leaf spot disease images by integrating local threshold and seeded region growing,” 2011 International Conference on Image Analysis and Signal Processing, 2011.

[5] V. Singh and A. Misra, “Detection of plant leaf diseases using image segmentation and soft computing techniques,” Information Processing in Agriculture, 2016.

[6] S. Prasad, S. K. Peddoju, and D. Ghosh, “Unsupervised resolution independent based natural plant leaf disease segmentation approach for mobile devices,” Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop on - I-CARE '13, 2013.

[7] M. G. Du and S. W. Zhang, “Crop Disease Leaf Image Segmentation Based on Genetic Algorithm and Maximum Entropy,” Applied Mechanics and Materials, vol. 713-715, pp. 1670–1674, 2015.

[8] B. Dhaygude & P.Kumbhar, “Agricultural plant Leaf Disease Detection Using Image Processing”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, no. 1, 2013, pp. 599-602.

[9] Z. H. Diao, Y. M. Song, H. Wang, and Y. P. Wang, “Study Surveys on Image Segmentation of Plant Disease Spot,” Advanced Materials Research, vol. 542-543, pp. 1047–1050, 2012.

[10] J. Y. Bai and H. E. Ren, “An Algorithm of Leaf Image Segmentation Based on Color Features,” Key Engineering Materials, vol. 474-476, pp. 846–851, 2011.

[11] J. Y. Bai and H. E. Ren, “An Algorithm of Leaf Image Segmentation Based on Color Features,” Key Engineering Materials, vol. 474-476, pp. 846–851, 2011.

[12] N. Valliammal and S. S.n.geethalakshmi, “A Novel Approach for Plant Leaf Image Segmentation using Fuzzy Clustering,” International Journal of Computer Applications, vol. 44, no. 13, pp. 10–20, 2012.

[13] L. Kaiyan, W. Junhui, C. Jie, and S. Huiping, “A Real Time Image Segmentation Approach for Crop Leaf,” 2013 Fifth International Conference on Measuring Technology and Mechatronics Automation, 2013.

[14] K. R. Gavhale, U. Gawande, and K. O. Hajari, “Unhealthy region of citrus leaf detection using image processing techniques,” International Conference for Convergence for Technology-2014, 2014.

[15] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610–621, 1973.

APPENDIX

Software and Hardware Requirement/Specification

There are software and hardware specification. Software is a conceptual entity which is a set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system. Hardware is component devices which are typically installed into or peripheral to a computer case to create a personal computer upon which system software is installed including a firmware interface.

Software Requirement

	Software	Purpose
Operating System	Microsoft Windows 8	As the operating system
	Microsoft Office Word 2010	For documentation
	Microsoft Power Point 2010	For slide presentation
	MATLAB R2016a	Developments tools
	Adobe Acrobat Reader	To read information from Internet

Hardware Requirement

	Component	Minimum Requirement
Laptop	Processor	Intel Core I3 processor
	RAM	4 GB
	CPU	64-bitoperatingsystem
Camera	Sensor	CMOS
	Zoom	7.78X
	Display	2.7-inchs
	Card Slot	SD/SDHC

INTRODUCTION TO MATLAB

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
- Data acquisition
- 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 non interactive 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 engines incorporate the LAPACK and BLAS libraries, embedding 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 add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learnand 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.

MATLAB (for matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++ and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems. In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises. In addition to the usage of MATLAB being integrated into the

teaching of Engineering and Linear Algebra courses, as part of their Continuing Studies programs, many Community Colleges and Universities are creating stand-alone MATLAB courses focused on just teaching the MATLAB user interface and script writing.

The MATLAB System

The MATLAB system consists of five main parts:

A. 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, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

B. The MATLAB Mathematical Function

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 eigen values, Bessel functions, and fast Fourier transforms.

C. 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.

D. Graphics

MATLAB has extensive facilities for displaying vectors and matGroundnuts as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image

processing, animation, and presentation graphics. It also includes low-level functions 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 includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-file.

MATLAB WORKING ENVIRONMENT

MATLAB DESKTOP

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are on the search path. The easiest way to see which directories are soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

This is a window where the output comments are displayed. It displays the output values and also errors if any. Various kinds of arithmetic operations can also be performed.

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB.The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

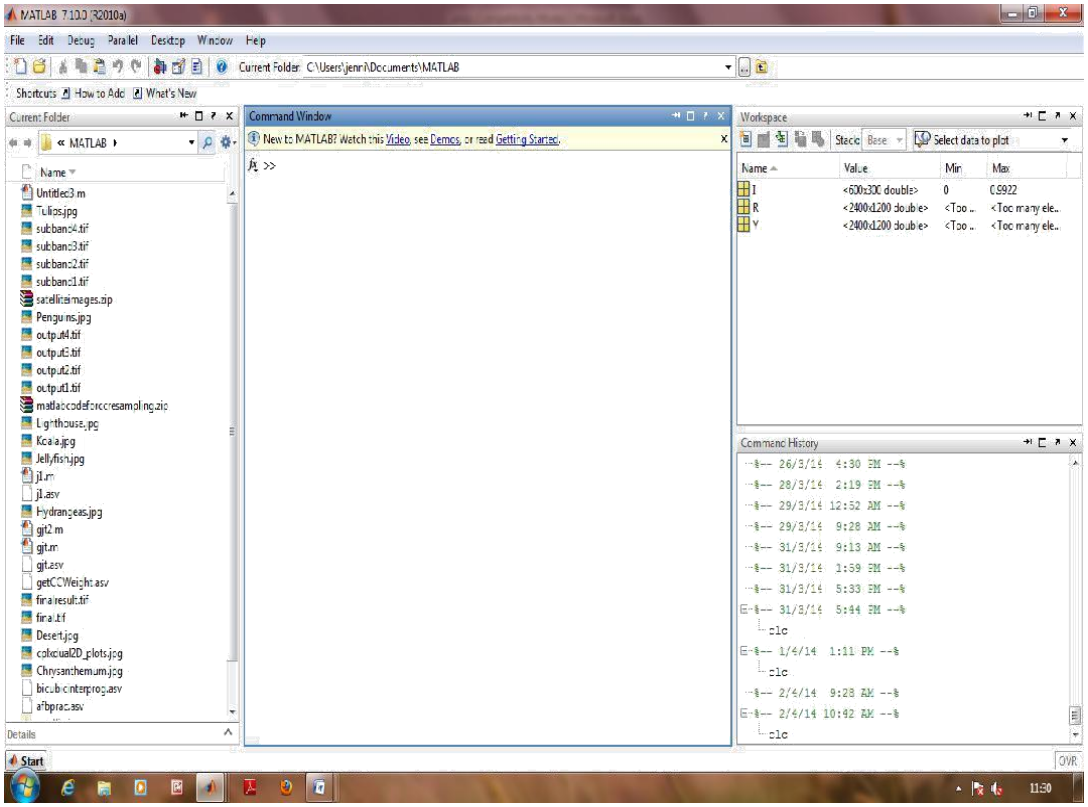


Fig: Default Window of MATLAB Software

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example, !emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favorites tab - View a list of documents you previously designated as favorites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dbstop, which sets a breakpoint.

If you just need to view the contents of an M-file, you can display it in the Command Window by using the type function.

MANIPULATING MATRICES

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices. Start MATLAB and follow along with each example.

You can enter matrices into MATLAB in several different ways:

- Enter an explicit list of elements.
- Load matrices from external data files.
- Generate matrices using built-in functions.
- Create matrices with your own functions in M-files.

Start by entering Dürer's matrix as a list of its elements. You have only to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon, ; , to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

To enter Dürer's matrix, simply type in the Command Window

A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

MATLAB displays the matrix you just entered.

A =

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

This exactly matches the numbers in the engraving. Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as A.

Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are:

- Variables
- Numbers
- Operators
- Functions

Variables

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

num_students = 25

Creates a 1-by-1 matrix named num_students and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. A and a are not the same variable. To view the matrix assigned to any variable, simply enter the variable name.

Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor.

Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

All numbers are stored internally using the long format specified by the IEEE floating-point standard. Floating-point numbers have a finite precision of roughly 16 significant decimal digits and a finite range of roughly 10-308 to 10+308.

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in Using MATLAB)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files.

You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, √-1
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2-\varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the callbacks, which are defined as subfunctions. This M-file is referred to as the application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

The following diagram illustrates the parts of a GUI implementation.

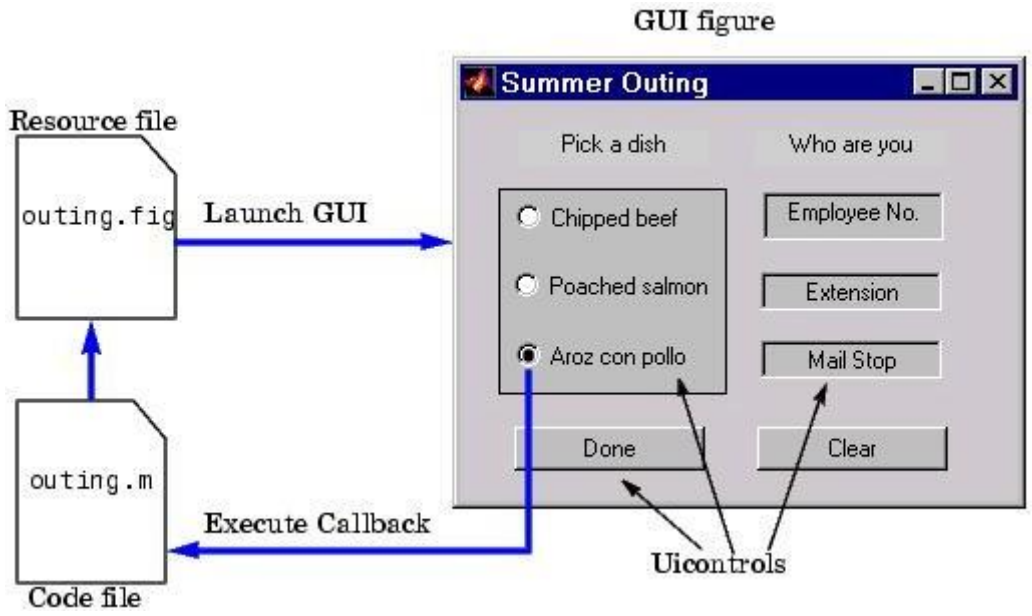


FIG 3.7.2 graphical user blocks

Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques used in the application M-file.

Application Examples - a collection of examples that illustrate techniques which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its non depressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback(h,eventdata,handles,varargin)
button_state = get(h,'Value');
if button_state == get(h,'Max')
   % toggle button is pressed
elseif button_state == get(h,'Min')
   % toggle button is not pressed
end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);
a(:,:,2) = rand(16,128);
a(:,:,3) = rand(16,128);
set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off)
set(off,'Value',0)
```

Obtaining the Radio Button Handles.

The handles of the radio buttons are available from the handles structure, which contains the handles of all components in the GUI. This structure is an input argument to all radio button callbacks.

The following code shows the call to mutual_exclude being made from the first radio button's callback in a group of four radio buttons.

```
function varargout = radiobutton1_Callback(h,eventdata,handles,varargin)
off = [handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];
mutual_exclude(off)
% Continue with callback
   .
   .
   .
```

After setting the radio buttons to the appropriate state, the callback can continue with its implementation-specific tasks.

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)
if (get(h,'Value') == get(h,'Max'))
   % then checkbox is checked-take approriate action
else
   % checkbox is not checked-take approriate action
end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)
user_string = get(h,'string');
% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)
user_entry = str2double(get(h,'string'));
if isnan(user_entry)
   errordlg('You must enter a numeric value','Bad Input','modal')
end
% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection.

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Triggering Callback Execution

MATLAB evaluates the list box's callback after the mouse button is released or a keypress event (including arrow keys) that changes the Value property (i.e., any time the user clicks on an item, but not when clicking on the list box scrollbar). This means the callback is executed after the first click of a double-click on a single item or when the user is making multiple selections. In these situations, you need to add another component, such as a Done button (push button) and program its callback routine to query the list box Value property (and possibly the figure Selection Type property) instead of creating a callback for the list box. If you are using the automatically generated application M-file option, you need to either: Set the list box Callback property to the empty string (') and remove the callback sub function from the application M-file. Leave the callback sub function stub in the application M-file so that no code executes when users click on list box items.

The first choice is best if you are sure you will not use the list box callback and you want to minimize the size and efficiency of the application M-file. However, if you think you may want to define a callback for the list box at some time, it is simpler to leave the callback stub in the M-file.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
switch val
case 1
% The user selected the first item
case 2
% The user selected the second item
% etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)
val = get(h,'Value');
string_list = get(h,'String');
selected_string = string_list{val}; % convert from cell array to string
% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is grayed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Axes

Axes enable your GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See Axes Properties for general information on axes objects.

Axes Callbacks

Axes are not uicontrol objects, but can be programmed to execute a callback when users click a mouse button in the axes. Use the axes ButtonDownFcn property to define the callback.

Plotting to Axes in GUIs

GUIs that contain axes should ensure the Command-line accessibility option in the Application Options dialog is set to Callback (the default). This enables you to issue plotting commands from callbacks without explicitly specifying the target axes.

GUIs with Multiple Axes

If a GUI has multiple axes, you should explicitly specify which axes you want to target when you issue plotting commands. You can do this using the axes command and the handles structure. For example,

```
axes(handles.axes1)
```

makes the axes whose Tag property is axes1 the current axes, and therefore the target for plotting commands. You can switch the current axes whenever you want to target a different axes. See GUI with Multiple Axes for and example that uses two axes.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.