ABSTRACT

Digital Image Processing is a rapidly evolving field with growing applications in Science and Engineering. Modern digital technology has made it possible to manipulate multi-dimensional signals. Digital Image Processing has a broad spectrum of applications. They include remote sensing data via satellite, medical image processing, radar, sonar and acoustic image processing and robotics.

This project propose an image pattern classification to identify various diseases in cattle with a combination of texture and color feature extraction. The purpose of this project is to find appropriate features that can identify cattle disease.

Firstly, normal and diseased images are collected and pre-processed. Then, features of shape, color and texture are extracted from these images. After that, these images are classified by support vector machine classifier. A combination of several features are used to evaluate the appropriate features to find distinctive features for identification of various cattle diseases. When a single feature is used, shape feature has the lowest accuracy hence shape feature is ignored and texture feature has the highest accuracy. A combination of texture and color feature extraction results a highest classification. A combination of texture and color feature extraction with polynomial kernel results best classification.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

The human visual system has no problem interpreting the subtle variations in translucency and shading in this Figure 1.1 photograph and correctly segmenting the object from its background.

Figure 1.1. Cattle as seen by the naked eye.

Agriculture is the major source of income in countryside areas. Animals like cow, buffalo, sheep, goat etc. play an important role in life of rural areas. They are used as a source of income. Hence animal husbandry becomes a most important concern. Lots of farmers are now torment from different killing diseases and increased breeding costs, etc. It is therefore essential for farmers to execute efficient and technical methods to increase productivity and the animal's husbandry. Most of the diseases now a days are affecting from domestic animals from anthrax to birds flew. Animal health is an important topic in recent days because of various viruses are affecting the human life are transferred from animal.

1.2 BACKGROUND:

Since recent decades, digital image processing, image analysis and machine vision have been sharply developed, and they have become a very important part of artificial intelligence and the interface between human and machine grounded theory and applied technology. These technologies have been applied widely in industry and medicine, but rarely in realm related to agriculture or natural habitats.

Despite the importance of the subject of identifying cattle diseases using digital image processing, and although this has been studied for at least 30 years, the advances achieved seem to be a little timid. Some facts lead to this conclusion

Methods are too specific. The ideal method would be able to identify any category of external diseases of cattle. Evidently, this is unfeasible given the current technological level. However, many of the methods that are being proposed not only are able to deal with only one kind of disease, but those diseases need to be at a certain stage in order to the algorithm to be effective. That is acceptable if the disease is in that specific stage, but it is very limiting otherwise. Many of the researchers do not state this kind of information explicitly, but if their training and test sets include only images of a certain stage, which is often the case, the validity of the results cannot be extended to other stages.

Operation conditions are too strict. Many images used to develop new methods are collected under very strict conditions of lighting, angle of capture, distance between object and capture device, among others. This is a common practice and is perfectly acceptable in the early stages of research. However, in most real world applications, those conditions are almost impossible to be enforced, especially if the analysis is expected to be carried out in a non-destructive way. Thus, it is a problem that many studies never get to the point of testing and upgrading the method to deal with more realistic conditions, because this limits their scope greatly. Lack of technical knowledge about more sophisticated technical tools. The simplest solution for a problem is usually the preferable one. In the case of image processing, some problems can be solved by using only morphological mathematical operations, which are easy to implement and understand. However, more complex problems often demand more sophisticated approaches. Techniques like neural networks, genetic algorithms and support vector machines can be very powerful if properly applied. Unfortunately, that is often not the case. In many cases, it seems that

the use of those techniques is in more demand in the scientific community than in their technical appropriateness with respect to the problem at hand. As a result, problems like over fitting, overtraining, undersized sample sets, sample sets with low representativeness, bias, among others, seem to be a widespread plague. Those problems, although easily identifiable by a knowledgeable individual on the topic, seem to go widely overlooked by the authors, probably due to the lack of knowledge about the tools they are employing. The result is a whole group of technically flawed solutions.

In recent times, computer vision methodologies and pattern recognition techniques have been applied towards automated procedures of disease recognition. Digital image processing is the use of the algorithms and procedures for operations such as image enhancement, image compression, image analysis, mapping, geo-referencing, etc. The influence and impact of digital images on modern society is tremendous and is considered as a critical component in variety of application areas including pattern recognition, computer vision, industrial automation and healthcare industries.

One of the most common methods in feature extraction is based on morphological features of image. Some simple geometrical features are aspect ratio, rectangularity, convexity, sphericity, form factor etc.

One can easily transfer the cattle's image to a computer and a computer can extracts features automatically in image processing techniques. Some systems employ descriptions used by veterinarians. But it is not easy to extract and transfer those features to a computer automatically.

The aim of the project is to develop a Disease recognition program based on specific characteristics extracted from photography. Hence this presents an approach where the disease is identified based on image features such as area, histogram equalization and edge detection and classification. The main purpose of this program is to use MATLAB resources.

Indeed, there are several advantages of combining MATLAB with the disease recognition program. The result proves this method to be a simple and an efficient attempt. Future sections will discuss more on image preprocessing and acquisition which includes the image preprocessing and enhancement, histogram equalization, edge detection. Further on sections introduces texture analysis and high frequency feature extraction of a cattle image to classify disease images i.e. parametric calculations and then followed by results.

CHAPTER 2

LITERATURE SURVEY

Agriculture is the major source of income in countryside areas. Animals like cow, buffalo, sheep, goat etc. play an important role in rural life. They not only contribute to their income but also their best insurance against any natural calamity. Hence animal husbandry becomes a most important concern. Lots of farmers are now torment from different killing diseases and increased breeding costs, etc. It is therefore essential for farmers to execute efficient and technical methods to increase productivity and reduce the animal husbandry The health of each animal on the farm is important, but livestock health is often discussed in terms of the whole herd. That's because whether there are two cattle or 200, what distresses one easily can affect the others.

Transmissible diseases, environmental conditions, and weather may affect every animal. Cattle management is the key to herd's health - how you move them, whether you quarantine sick animals, grazing methods, parasite control, and much more. Well-planned herd management, which includes minimizing stress and crowding, careful care of calves, and implementing a good vaccination program, can help prevent most illnesses. The total cattle contributes around37.28% of the total livestock population, the total number of cattle in the country as per 2015 census is 190 million.

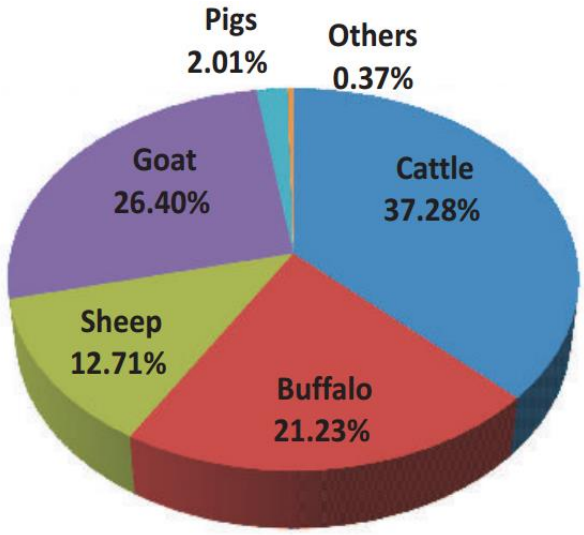


Figure 2.1 Pie chart for total distribution of the available livestock in India

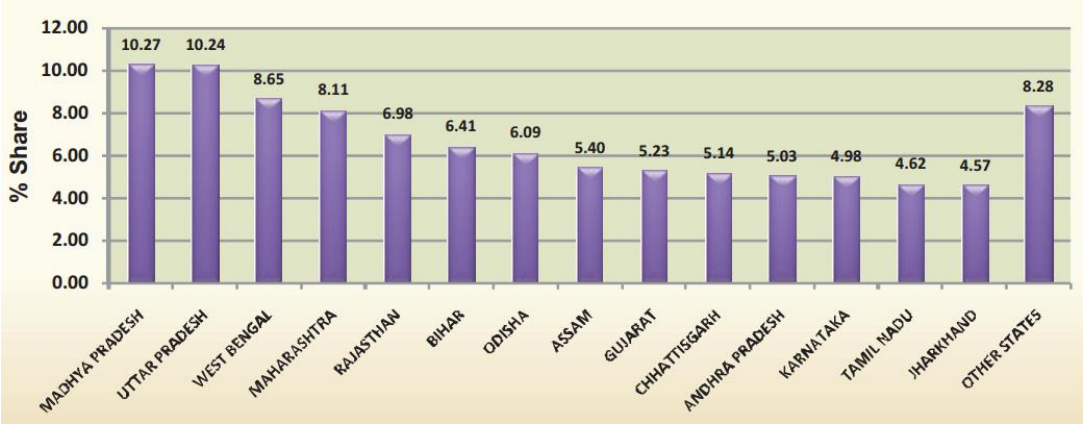


Figure 2.2 State wide distribution of the total available cattle population

Cattle suffer from a different variety of diseases, which may be categorized into various categories such as hereditary, congenital, acquired diseases, contagious, non-contagious and diseases due to viruses and parasites.

In places with high cattle population such as dairy farms, day to day monitoring of cattle health is difficult. Diseases such as Foot and Mouth , Udder inflammation, parasitic skin diseases, and other externally visible diseases cannot be monitored on a day to day basis, which may later lead to death of cattle due to lack of initial treatment. Hence a system which can monitor such initial conditions/symptoms has to be developed.

2.1 Existing Techniques:

2.1.1 Wireless Sensor Network for Cattle Monitoring System:

Wireless Sensor Network (WSN) technology for monitoring the health of dairy cows. By monitoring and understanding the cow individual and herd behavior, farmers can potentially identify the onset of illness, lameness or other undesirable health conditions. However, the WSN implementation needs to cope with various technical challenges before it can be suitably and routinely applied in cow management. This technique discusses results concerning data transportation (i.e. mobility) from the cow mounted sensory devices.

2.2 Early Detection of External Diseases Using Support Vector Machine (SVM):

This project deals with a new type of early detection of diseases. Images of the cattle's affected by diseases are acquired by using a digital camera. The images are processed for getting a gray colored image and then using image segmentation, image classification techniques to detect the particular disease. The image is transferred to the analysis algorithm to report the quality. The technique evolved in this system is both image processing and soft computing. The image processing technique is used to detect the disease and soft computing technique is used for doing this detection over a wide population. The images are acquired by using a digital camera of approximately 12 M-Pixel resolution in 24-bits color resolution. The images are then transferred to a PC and represented in MATLAB software. The RGB image is then segmented using blob like algorithm for segmentation of inflammations/diseases on cattle body.

The segmented body part is now analyzed for detecting disease. The Support Vector Machine classifier is used to classify the disease types. It is also implemented in FPGA kit by converting the MATLAB coding into HDL coder. In FPGA, the input image is downloaded to the memory. It reads the image from memory, process it and display the output image on monitor. A software routine is written in MATLAB. In which training and testing performed via several neural network classifier. Texture Feature Classification Methods are as follows.

2.2.1. K-nearest neighbor:

K-nearest neighbor classifier is used to calculate the minimum distance between the given point and other points to determine the given point belongs to which class. Goal is to computes the distance from the query sample to every training sample and selects the neighbor that is having minimum distance.

2.2.2. Radial basis function:

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin. The normally used measuring norm is Euclidean distance. RBF's are the networks where the activation of hidden units is based on the distance between the input vector and a prototype vector.

2.2.3. Artificial neural networks (ANNs):

ANNs are popular machine learning algorithms that are in a wide use in recent years. Multilayer Perception (MLP) is the basic form of ANN that updates the weights through back propagation during the training. There are other variations in neural networks, which are recently, became popular in texture classification Probabilistic Neural Network (PNN): It is derived from Radial Basis Function (RBF) network and it has parallel distributed processor that has a natural tendency for storing experiential knowledge. PNN is an implementation of a statistical algorithm called kernel discriminate analysis in which the operations are organized into a multilayered feed forward network having four layers viz. input layer, pattern layer, summation layer, and output layer.

2.3 Back propagation network:

A typical Back Propagation network (BP) consists of three parts: input layer, hidden layer and output layer. Three parts in turn connect through the collection weight value between nodes. The largest characteristic of BP network is that network weight value reach expectations through the sum of error squares between the network output and the sample output, and then it continuously adjusted network structure's weight value. It is popular and extensively used for training feed forward networks. Also it has no inherent novelty detection, so it must be trained on known outcomes for training feed forward networks.

2.4 Support vector machine (SVM):

SVM is a non-linear classifier, and is a newer trend in machine learning algorithm. SVM is popularly used in many pattern recognition problems including texture classification. SVM is designed to work with only two classes. This is done by maximizing the margin from the hyperplane. The samples closest to the margin that were selected to determine the hyperplane is known as support vectors. Multiclass classification is applicable and basically built up by various two class SVMs to solve the problem, either by using one-versus-all or one. Another feature is the kernel function that projects the non-linearly separable data from low-dimensional space to a space of higher dimension so that they may become separable in the higher dimensional space too.

The first step in the proposed approach is to capture the sample from the digital camera and extract the features. The sample is captured from the digital camera and the features are then stored in the database. Preprocessing images is used to removing low-frequency background noise. Normalizing the intensity of the individual particles of images. It enhance the visual appearance of images. Improve the manipulation of datasets. It is the technique of enhancing data images prior to computational processing. The caution is enhancement techniques can emphasize image artifacts, or even lead to a loss of information if not correctly used. The steps involved in preprocessing are to get an input image and then the image has to be enhanced.

Then the RGB image is converted to a gray scale image to get a clear identification of external diseases in cattle. Noise removal function can be performed by using filtering techniques. Mean filtering: The 3x3 sub-region is scanned over the entire image. At each position the center pixel is replaced by the average value. Median filtering: The 3x3 sub-region is scanned over the entire image. At each position the center pixel is replaced by the median value. The PSNR value is calculated for both the mean and median filter. Based on the PSNR value one of the filtering image is taken for a further process. For mean filtering, the PSNR value is 23.78 and the PSNR value for median filtering is 12.89. The higher the PSNR, the better the quality of the compressed or reconstructed image. Therefore the mean filtering is taken for the further process.

Image segmentation in general is defined as a process of partitioning an image into homogenous groups such that each region is homogenous but the union of no two adjacent regions is homogenous [11].Image segmentation is performed to separate the different regions with special significance in the image. These regions do not intersect each other. Blob detection helps to obtain Regions of Interest for further processing. It is applied for the presence of same type of objects in multiples. Segment the objects of interest from the complex background.

Image features usually include color, shape and texture features. Feature extraction is performed related to the Majority Based Voting method there are 3steps involved: 1) Histogram Oriented Gradient (HOG), 2) Gaussian Mixture Model (GMM) and 3) Gabor Feature. HOG is the feature descriptors used for the purpose of object detection. Gaussian mixture model is used for the texture analysis. Gabor Feature is calculate the relationship between groups of two pixels in the original image. In this proposed work, the image can be sub divided into small block. Then in each block the three steps are involved. HOG is used for detecting the distribution of color ratio in

an image. GMM used for the detection of shape of inflammations present in an image. Gabor feature can be used to find the orientation of inflammations. Finally, the feature values are fed as input to the classifiers.

The back propagation and feed forward classifiers are not detecting some inflammations in an image. But SVM gives better result. SVM is a non-linear classifier, and is a newer trend in machine learning algorithm. SVM is popularly used in many pattern recognition problems including texture classification. SVM is designed to work with only two classes. This is done by maximizing the margin from the hyperplane. The samples closest to the margin that were selected to determine the hyperplane is known as support vectors [12]. Multiclass classification is applicable and basically built up by various two class SVMs to solve the problem, either by using one-versus-all or one. Another feature is the kernel function that projects the non-linearly separable data from low-dimensional space to a space of higher dimension so that they may become separable in the higher dimensional space too. It is used to detect the inflammations and also gives information about a type of diseases. It gives a result of number of pests are presented. Then, it gives a remedy to take over for controlling a disease. Finally, the feature values are fed as input to the Support Vector Machine classifier, allow us to accurately distinguish the unhealthy and healthy cattle. This is an important step towards the identification of diseases and to take the corresponding remedies.

2.5 Classification of common diseases of cattle and their symptoms:

2.5.1 Viral Diseases:

Sl.no	Disease	Symptoms
1	Cow Pox	Fever accompanied by appearance of small nodules.
2	Rinder Pest	Constipation followed by severe diarrhea.
3	Foot and Mouth	Blisters appear on the moth and foot resulting in extreme soreness of the parts. Loss of appetite, excessive salivation, high fever accompanied by shivering, inability to work.
4	Rabies	Marked changes in behavior, restlessness and paralysis (Symptoms appear in 14 – 90 days).
5	Dermatitis	Irritation, Blisters and eruptions on the skin surface.

2.5.2 Bacterial diseases:

Sl.no	Disease	Symptoms
1	Anthrax	Fever with swelling of body, milk secretion reduced.
2	Haemorrhagic Septicamia	High fever, anorexia, increased respiration, marked salivation.
3	Blackquarter	A fatal toxemia especially of young cattle.
4	Tuberculosis	Fever, infection of udder, lungs, intestines, and other parts.
5	Brucellosis	Sterility due to infection in the reproductive organs, e.g., uterus in females and testes in males.
6	Mastitis	Fever, udders become swollen, milk is watery.
7	Salmonellosis	Fever, diarrhea with blood clots.

2.5.3 Fungal Disease:

Sl.no	Disease	Symptoms
1	Ringworm	Small, circular, discolored raised patches.

2.5.4 Parasitic Disease:

Sl.no	Disease	Symptoms
1	External parasites(Lice, ticks)	Live on skin and cause skin disease.
2	Internal parasites(Worms,Fluke)	Live in stomach, intestines and damages the liver of cattle.

2.6 Foot and Mouth Disease:

Foot and Mouth Disease (FMD) is a severe, highly contagious viral disease of livestock with significant economic impact. The disease affects cattle and swine as well as sheep, goats, and other cloven-hoofed ruminants. All species of deer and antelope as well as elephant, and giraffe

are susceptible to FMD. FMD is characterized by fever and blister-like sores on the tongue and lips, in the mouth, on the teats and between the hooves. The disease causes severe production losses and while the majority of affected animals recover, the disease often leaves them weakened and debilitated. The organism which causes FMD is an aphthovirus of the family Picornaviridae. There are seven strains (A, O, C, SAT1, SAT2, SAT3, Asia1) each one requiring a specific vaccine strain to provide immunity to a vaccinated animal.

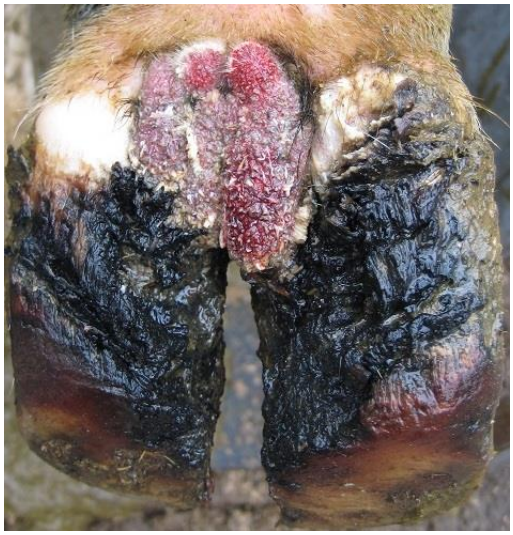


Figure 2.3 Foot and Mouth Disease (FMD)

2.6.1 Clinical signs:

The severity of clinical signs will depend on the strain of virus, the age and species of animal. The signs can range from a mild infection to severe. Clinical signs are more severe in cattle. The typical clinical sign is the occurrence of blisters (or vesicles) on the nose, tongue, lips, oral cavity, between the toes, above the hooves, teats and pressure points on the skin. Ruptured blisters can result in extreme lameness and reluctance to move or eat. Secondary bacterial infection of open blisters can also occur. Other symptoms often seen are fever, depression, hyper salivation, loss of appetite and weight, drop in milk production

2.7 Mastitis:

Mastitis in dairy cattle is the persistent, inflammatory reaction of the udder tissue. Mastitis, a potentially fatal mammary gland infection, is the most common disease in dairy cattle in the United

States. It is also the most costly to the dairy industry. Milk from cows suffering from mastitis has an increases somatic cell count. Mastitis occurs when white blood cells (leukocytes) are released into the mammary gland, usually in response to bacteria invading the teat canal. Milk-secreting tissue and various ducts throughout the mammary gland are damaged due to toxins released by the bacteria. Mastitis can also occur as a result of chemical, mechanical, or thermal injury. This disease can be identified by abnormalities in the udder such as swelling, heat, redness, hardness, or pain (if it is clinical). Other indications of mastitis may be abnormalities in milk such as a watery appearance, flakes, or clots. When infected with subclinical mastitis, a cow does not show any visible signs of infection or abnormalities.

Figure 2.4 Cattle with Mastitis disease

2.7.1 Clinical signs:

Clinical mastitis, Sub-Clinical mastitis, per acute mastitis, acute mastitis, sub-acute mastitis, chronic mastitis. Mastitis can cause a decline in potassium and lactoferrin. It also results in decreased casein, the major protein in milk. As most calcium in milk is associated with casein, the disruption of casein synthesis contributes to lowered calcium in milk. The milk protein continues to undergo further deterioration during processing and storage. Milk from cows with mastitis also has a higher somatic cell count. Generally speaking, higher the somatic cell count, the lower the milk quality.

2.8 Ringworm:

This fungal skin disease often appears in calves and yearlings during winter, but generally disappears without treatment by spring. There are several species of Trichophyton and Microsporum fungi that cause ringworm. The fungal spores may be spread by contact with an animal that has ringworm or by spores in the environment, something an infected animal has rubbed on. Spores can be spread by equipment used on an infected animal and then a susceptible one such as a rope or halter, grooming tool, or equipment used on more than one animal. The fungus becomes established on the skin of the susceptible animal and infiltrates hair follicles. Lesions develop about three weeks later.

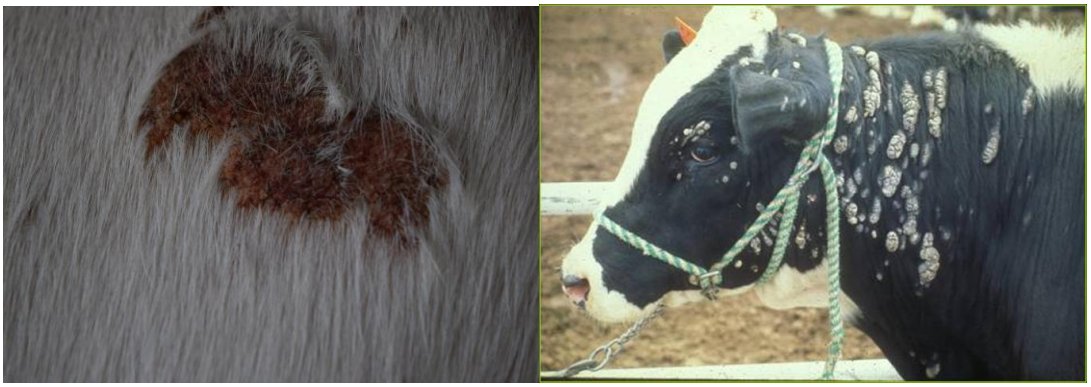


Figure 2.5 Ringworm disease on Cattle's skin

2.8.1 Clinical symptoms:

In the early stages, affected areas are small, with raised skin and rough hair. After several weeks the hair falls out, leaving thickened patches of scaly gray lesions, often on the face, neck and around the eyes.

2.9 Pink Eye Disease:

Pinkeye is a painful, debilitating condition that can severely affect animal productivity. Pink eye is caused by the bacterium MORAXELLA BOVIS, which infects the eye and produces a toxin. The toxin attacks the surface of the eye (cornea and the surrounding membranes (conjunctivae), eroding the surface and causing severe inflammation and, in severe cases, temporary or permanent blindness.

Important factors that pred include:

• Dusty conditions

• Flies, bright sunlight and physical irritation of the eye

2.9.1 Clinical symptoms:

Stage 1: The first sign of pinkeye is an animal with a 'runny eye'. In the first two days, the membranes of the eye are red and swollen

Stage 2: the spot in the Centre of the eye continues to enlarge. Over the next one to two weeks the cornea is eroded to form an ulcer that spreads and swells, with most of the eye changing from white to yellow and then to red

Stage 3: Most of the eye becomes red as blood vessels grow across the cornea. As recovery progresses, the blood vessels start to recede and the eye first becomes a cloudy blue colour, then begins to clear. Recovery is usually complete 3–5 weeks after the initial infection

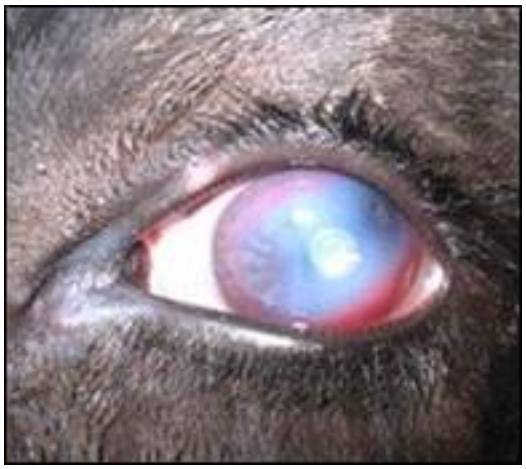


Figure 2.6 Cattle infected by Pink Eye disease

2.10 Classification of Cattle Diseases using Color Texture Features:

This paper describes Support Vector Machine (SVM) and Artificial Neural Network (ANN) based recognition and classification of visual symptoms of the disease. Color images of diseased regions of different affected cattle's are used in this work. Different types of symptoms affected by fungal diseases namely ringworm, viral diseases like Foot and Mouth disease, Mastitis

are considered for the study. The developed algorithms are used to preprocess, segment, extract features from disease affected regions. The affected regions are segmented using k-means segmentation technique. Color texture features are extracted from affected regions and then used as inputs to SVM and ANN classifiers. The texture analysis is done using Color Co-occurrence Matrix. Tests are performed to classify image samples. Classification accuracies between 68.5% and 87% are obtained using ANN classifier. The average classification accuracies have increased to 77.5% and 91.16% using SVM classifier. The detailed block diagram of adopted methodology is shown in Figure 2.7

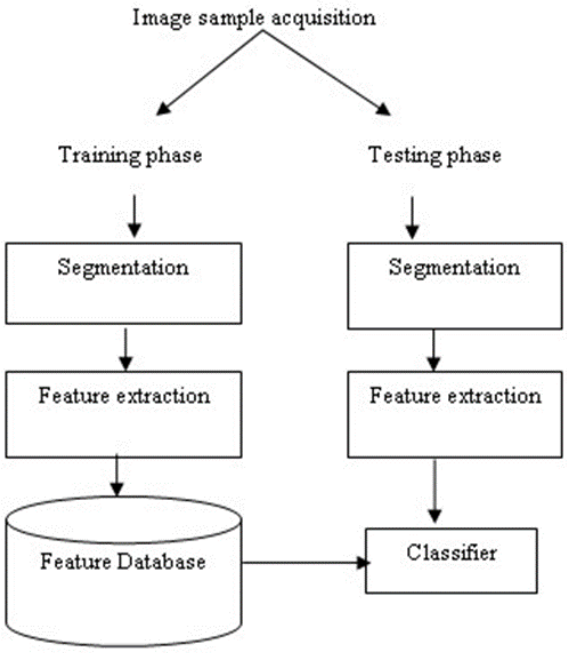


Figure 2.7 Proposed Block Diagram for Classification of Diseases

Single image is captured by analog camera. Then preprocessing steps are applied over image. The preprocessing of image includes shade correction, removing artifacts, formatting. Formatting deals with storage representation and setting the attributes of the image. In computer vision, segmentation refers to the process of clustering the pixels with certain properties into salient regions and these regions correspond to individual surfaces, objects or natural parts of the objects. We proposed k-means segmentation technique to segment target regions. Target regions are those areas in the image that represented visual symptoms of a disease. The image analysis technique is done using Color Co-occurrence Matrix (CCM). The inflammated/affected areas vary in color and texture and are dominant in classifying diseases symptoms. So, we have considered both color and

texture features for recognition and classification purpose. Image texture, defined as a function of the spatial variation in pixel intensities (gray values). The use of color features in the visible light spectrum provided additional image characteristic features over traditional gray-scale representation. CCM is a method in which both color and texture features are taken into account to arrive at unique features which represent that image. The CCM method involved three major steps. First, transformations of an RGB (Red, Green, and Blue) color representation of an image to an equivalent HSI (Hue, Saturation and Intensity) color representation. Once this process is completed Color Co-occurrence Matrices from the HSI pixels is generated. Lastly, texture features from the CCM matrices are generated [8].

Each pixel map is used to generate a Color Co-occurrence Matrix, resulting in three CCM matrices, one for each of the H, S and I pixel maps. The Color Co-occurrence texture analysis method was developed through the use of spatial gray level dependence matrices (SGDM's). The Gray Level Co-occurrence Method (GLCM) is a statistical way to describe shape by statistically sampling the way certain grey-levels occur in relation to other grey-levels. The GLCM is based on the repeated occurrence of some gray-level configuration in the texture. This method measures occurrence of gray levels between a specific position P (i, j) in the image and a neighboring pixel, according to a given distance d and direction θ. The CCM matrices are then normalized.

Support Vector Machine comprise of a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an n-dimensional space, SVM constructs a separating hyperplane in the space, one which maximizing the margin between the given two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes, since in general the larger the margin the better the generalization error of the classifier. Classifying data is a common need in machine learning. Suppose some given data points belong to one of two classes and the goal is to decide, which class a new data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p−1) dimensional hyperplane. This is called a linear classifier. The classifier is also known as a maximum margin classifier.

2.11 Artificial Neural Network:

An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Multilayer feed-forward neural networks are the most commonly used neural networks for object identification and classification. The layers of neurons between the inputs and the "output layer" are called "hidden layer". Back propagation neural network (BPNN) is the most important algorithm for the supervised training of multilayer feed-forward ANN.

The BPNN are simple and effective to implement and found suitable for a wide range of machine learning applications. The number of neurons in the input layer corresponds to the number of input features and the number of neurons in the output layer corresponds to the number of classes. The number of nodes in the hidden layer is calculated.

All the algorithms used in this work are implemented in MATLAB. In this work, we have used the OSU-SVM toolbox available in MATLAB 7.0. The core of this toolbox is based on Dr. Lin's Lib SVM version 2.33. It is developed by Junshui Ma, Los Alamos National Lab and Yi Zhao, EE department, Ohio State University. For classification using Neural Networks, the Neural Network Toolbox available in MATLAB 7.0 was used. The SVM and ANN are trained with 50 images of each type. The remaining 75 images are used for testing. Around 15% image samples are used for validation of the designed classifier model.

The technology leverage farmers can look at the possibility of diseases at early stages take decision on possible treatment, and the like. The identification of the symptoms affected by infections and inflammations, by means of a machine vision system may support farmers in proper assessment of cattle.

Here we used image samples of cattle's that showed visual symptoms of a disease. These diseased regions were identified and segmented using k-means segmentation. Color texture features were extracted from each segmented region and used as inputs to a SVM and ANN classifiers. The performance of SVM classifier found to be better than ANN classifier for the work done.

CHAPTER 3

CATTLE DISEASE DETECTION USING SVM

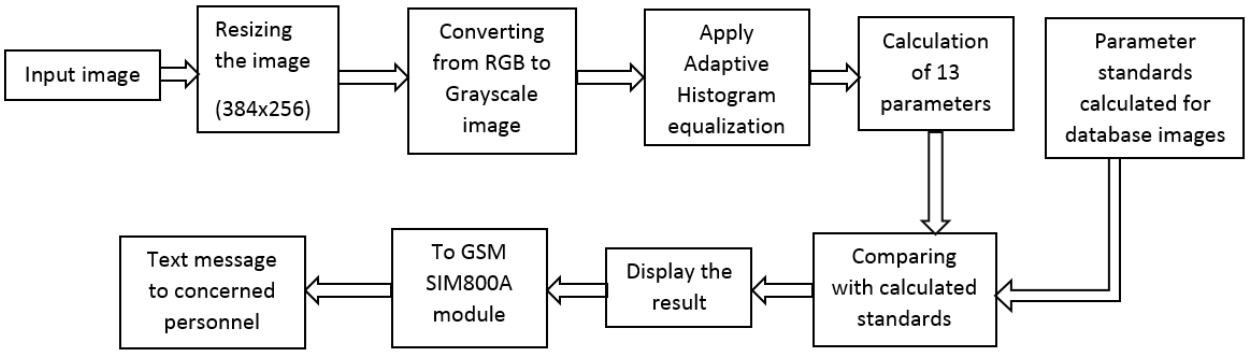


Figure 3.1 Block Diagram

3.1 MATLAB:

In this introduction we will describe how MATLAB handles simple numerical expressions and mathematical formulas.The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment. It has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

3.2 Starting MATLAB

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 8.1.2) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

- The Command History
- The Workspace
- The Current Directory
- The Help Browser
- The Start button

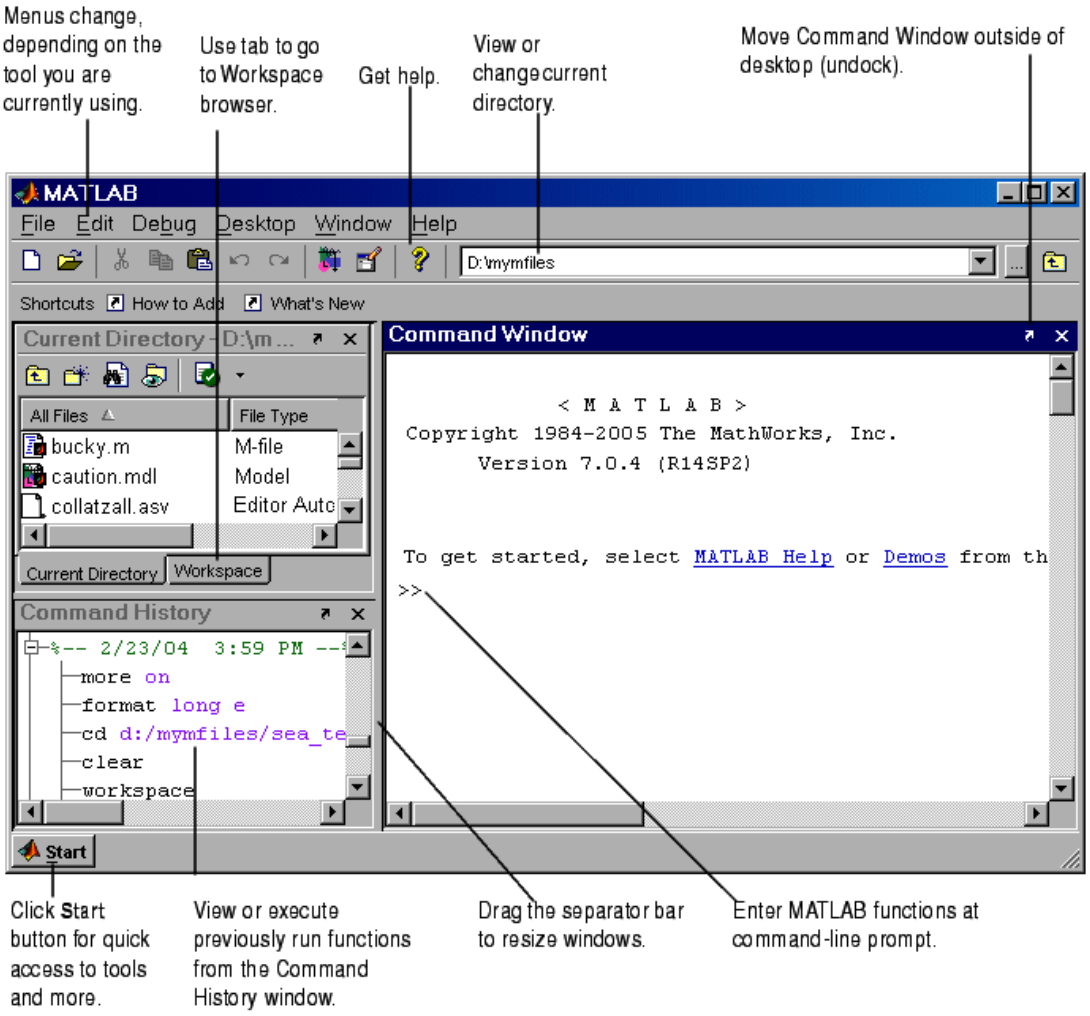


Figure 3.2: The graphical interface to the MATLAB workspace

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure 3.2. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs.

Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run.

You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

>> for full version

EDU> for educational version

Note: To simplify the notation, we will use this prompt, >>, as a standard prompt sign, though our MATLAB version is for educational purpose.

Using MATLAB as a calculator

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate the expression, 1 + 2 × 3. You type it at the prompt command (>>) as follows,

```
>> 1+2*3
ans = 7
```

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable ans, short for answer, to store the results of the current calculation. Note that the variable ans is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example,

```
>> x = 1+2*3
x = 7
```

Will result in x being given the value 1 + 2 × 3 = 7. This variable name can always be used to refer to the results of the previous computations. Therefore, computing 4x will result in

```
>> 4*x
ans = 28.0000
```

3.3 HARDWARE REQUIREMENTS:

3.3.1 GSM SIM800A MODULE:

SIM800a is a quad-band GSM/GPRS module that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800 features GPRS multi-slot class 12/ class 10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.With a tiny configuration of 24*24*3mm, SIM800 can meet almost all the space requirements in user applications, such as M2M, smart phone, PDA and other mobile devices.

SIM800 has 68 SMT pads, and provides all hardware interfaces between the module and customers' boards.

- Support up to 5*5*2 Keypads.
- One full function UART port, and can be configured to two independent serial ports.
- One USB port can be used as debugging and firmware upgrading.
- Audio channels which include a microphone input and a receiver output.
- Programmable general purpose input and output.
- TTL Rx and TTL Tx and DB9 Connector Based RS232 Output
- Voltage Supply Required- 9VDC to 12VDC with at least 2A Peak Current Capability
- One SIM card interface.
- Support Bluetooth function.
- Support one PWM.
- Operating temperature: -40C to +85C
- PCM/SPI/SD card interface, only one function can be accessed synchronously. (default is PCM)
- SIM800 is designed with power saving technique so that the current consumption is as low as 1.2mA in sleep mode.
- SIM800 integrates TCP/IP protocol and extended TCP/IP AT commands which are very useful for data transfer applications.

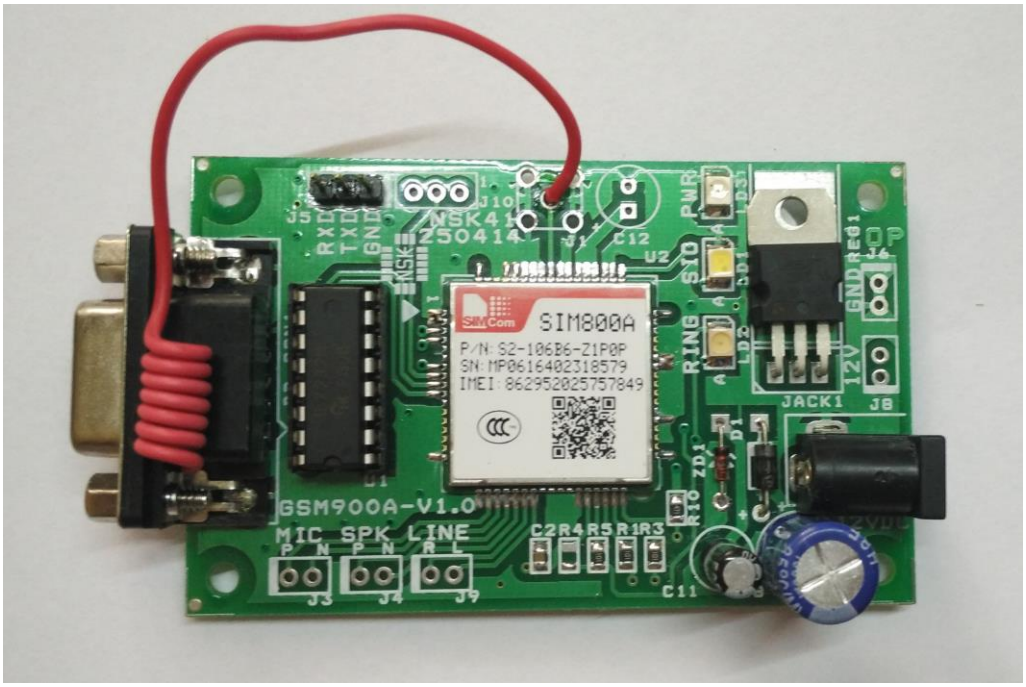


Figure 3.3 GSM SIM800A MODULE

3.3.2 USB TO UART CONVERTER:

This is a USB to Serial UART (TTL level) converter module. It is allow you to connect your computer through USB port and use it as a regular serial communication. All USB protocol is handled within the module. There is no other device or programming required. This board features innovations that set it apart from other USB to Serial Converter boards. Innovations feature like 256 byte receive buffer and 128 byte transmit buffer utilize new buffer smoothing technology to allow for high data throughput. Also, 6MHz clock output signal options for driving external MCU or FPGA.

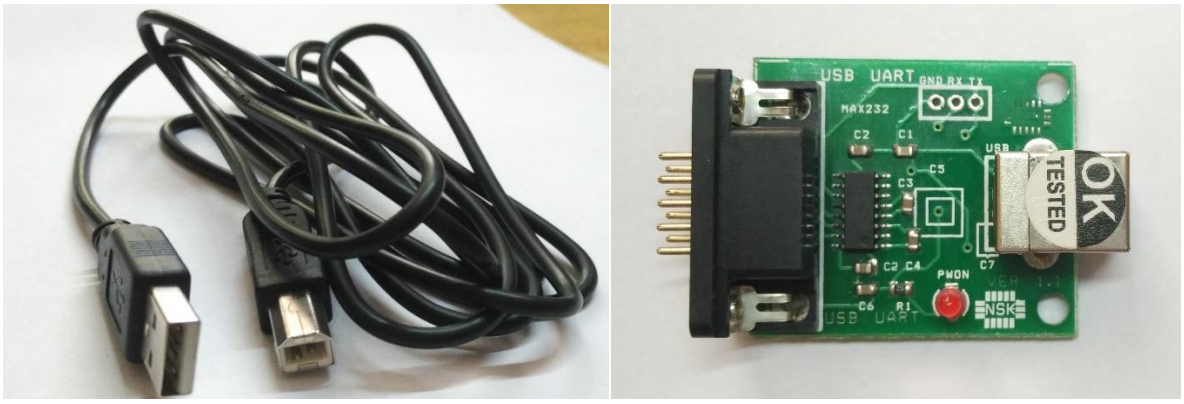
The TX and RX pins from the USB-SER can be connected directly to RX and TX pins of your preferred microcontroller or serial application for a simple serial cable replacement connection. The USB-SER board is perfect for embedded systems that require a serial connection to a computer. The board attaches directly to the USB bus via a standard type mini B receptacle connector. It shows up on any Windows computer as a standard serial COM port. Any applications that talk to this COM port is automatically convert to USB and back to UART to your target board.

TX: Transmit from the USB-SER board. It is a Serial Output and normally connected to the RX pin on any microcontroller or equivalent UART.

RX: Receive into the USB-SER board. It is a Serial Input and normally connected to the TX pin on any microcontroller or equivalent UART.

+5V: is connected to USB 5V bus via filter ferrite bead. It can range from 4.0 – 5.0 V depending on the regulation of the given USB port. In USB 2.0 system, this pin can provide a maximum of 500mA in accordance with the USB 2.0 specification.

- SiLabs CP 2102 based
- Self-Powered through USB
- 9Pin D type male Connecter for RS232 output



(a) **(b)**

Figure 3.4 USB to UART (a) Connecting cable (b) Converter

3.3.3 ADAPTER:

A power supply that is built into an AC mains power plug is known as a "plug pack" or "plug-in adapter", or by slang terms such as "wall wart". They are even more diverse than their names; often with either the same kind of DC plug offering different voltage or polarity, or a different plug offering the same voltage. "Universal" adapters attempt to replace missing or damaged ones, using multiple plugs and selectors for different voltages and polarities.

Replacement power supplies must match the voltage , and supply at least as much current as, the original power supply.

The least expensive AC units consist only of a small transformer, while DC adapters include a few additional diodes. Whether or not a load is connected to the power adapter, the transformer has a magnetic field continuously present and normally cannot be completely turned off unless unplugged. Because they consume standby power, they are sometimes known as "electricity vampires" and may be plugged into a power strip to allow turning them off.

Figure 3.5 12V-2A Adapter

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Flow Diagram:

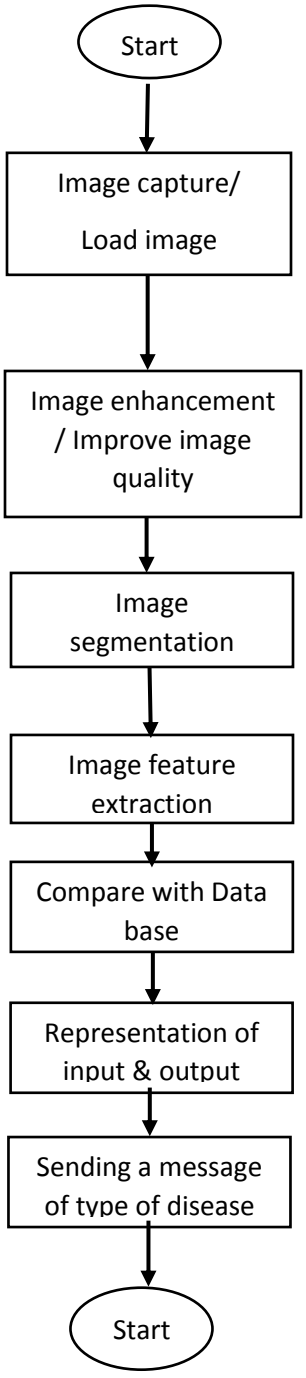


Figure 4.1 Flow chart

4.2 Texture Feature Extraction:

Gray Level Co-occurrence Matrix (GLCM) extract second order statistical texture features. Texture feature extraction used in this research are contrast, correlation, energy and homogeneity. This features taken from research [3] to extract texture feature in diseased region.

Contrast of the pixel and its neighbors is calculated over all of the image pixels. Contrast is used to measure contrast between neighborhood pixels.

$$f1 = \sum_{i=1}^{k} \sum_{j=1}^{k} (i-j)^2 Pij \qquad (1)$$

Correlation is a measure of correlation of a pixel with its neighbors over all of the image.

$$f2 = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{(i-mr)(j-mc)Pij}{\sigma_r \sigma_c} \qquad (2)$$

Energy is a sum of G (grey level co-occurrence matrix elements.

$$f3 = \sum_{i=1}^{k} \sum_{j=1}^{k} Pij^2 \qquad (3)$$

Homogeneity computes similarity of G to the diagonal matrix.

$$f4 = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{Pij}{1+|i-j|} \qquad (4)$$

All of the four features [10] described in this section represent texture of the images of diseased region in comparison with the normal one.

4.3 Color Feature Extraction:

Color is a distinctive feature for image representation that is invariant with respect to scaling, translation and rotation of an image [9]. Mean, skewness and kurtosis are used to represent color as features. To do this, we transform RGB to LAB.

Mean used to represent average value of each color channel [11].

$$\mu = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} Pij \qquad (5)$$

Skewness and kurtosis used to measure the distribution of each color channel. Skewness can be described as:

$$s = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(Pij-\mu)^3}{MN\sigma^3} \qquad (6)$$

Skewness is a measure of symmetry. If a distribution or data is symmetric, it looks the same to the left and right of the center point. Kurtosis can represent whether the data are peaked or flat relative to a normal distribution [12]. Kurtosis can be described as follows:

$$k = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(Pij-\mu)^4}{MN\sigma^4} \qquad (7)$$

Combination of mean, skewness and kurtosis is used to represent color feature of normal and diseased image of leaf.

4.4 Support Vector Machine (SVM):

Training sample in support vector machine is separable by a hyperplane. This hyperplane is computed according to the decision function f(x) = sign (w.x) + b, where w is a weight vector and b is a threshold cut-off.

To maximize the margin, w € f and b have to be minimized to:

$$\begin{aligned} Xi.w + b &\geq +1 \; for \; yi = +1 \\ Xi.w + b &\leq -1 \; for \; yi = -1 \\ Yi\,(xi.w + b) - 1 &\geq 0 \end{aligned} \qquad (8)$$

Additional slack variables should be added to prevent over fitting.

$$\begin{aligned} Xi.w + b &\geq +1 - \xi \; for \; yi = +1 \\ Xi.w + b &\leq -1 + \xi \; for \; yi = -1 \\ Yi\,(xi.w + b) - 1 + \xi &\geq 0 \end{aligned} \qquad (9)$$

SVM was chosen as the binary classifier because it can classify accurately even when limit samples were available [6]. There are different kernel function in SVM classifier. Table 1 shows different types of kernel in SVM. They are linear, quadratic, polynomial, and radial basis function.

Table 1 Different kernel in SVM

linear	$K(x_i, x_j) = x_i^T x_j$
quadratic	$K(x_i, x_j) = ((x_i, x_j^T)+1))^2$
polynomial	$K(x_i, x_j) = ((x_i, x_j^T)+1))^3$
Radial basis function	$K(x_i, x_j) = \exp\left(-\frac{\|\|xi-xj^T\|\|^2}{2\sigma^2}\right)$

SVMs (Support Vector Machines) are a useful technique for data classification. Classification task usually involves separating data into training and testing sets. Each instance in the training set contains one \target value" (i.e. the class labels) and several attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. Let's consider the following simple problem:

We are given a set of n points (vectors) : x1, x2, x3, ….. xn such that xi is a vector of length m, and each belong to one of two classes we label them by "+1" and "-1". So our training set is

$$(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$$

$$\forall i \;\; x_i \in R^m, y_i \in \{+1, -1\} \qquad (10)$$

We want to find a separating hyperplane that separates these points into the two classes. "The positives" (class "+1") and "The negatives" (class "-1"). Let's introduce the notation used to define formally a hyperplane:

$$f(x) = \beta_0 + \beta^T x,$$

$$\beta_0 \qquad (11)$$

Where, β is known as the *weight vector* and βo as the *bias*. For a linearly separable set of 2D points which belong to one of two classes, find a separating straight line.

In Figure 7.1 you can see that there exist multiple lines that offer a solution to the problem.

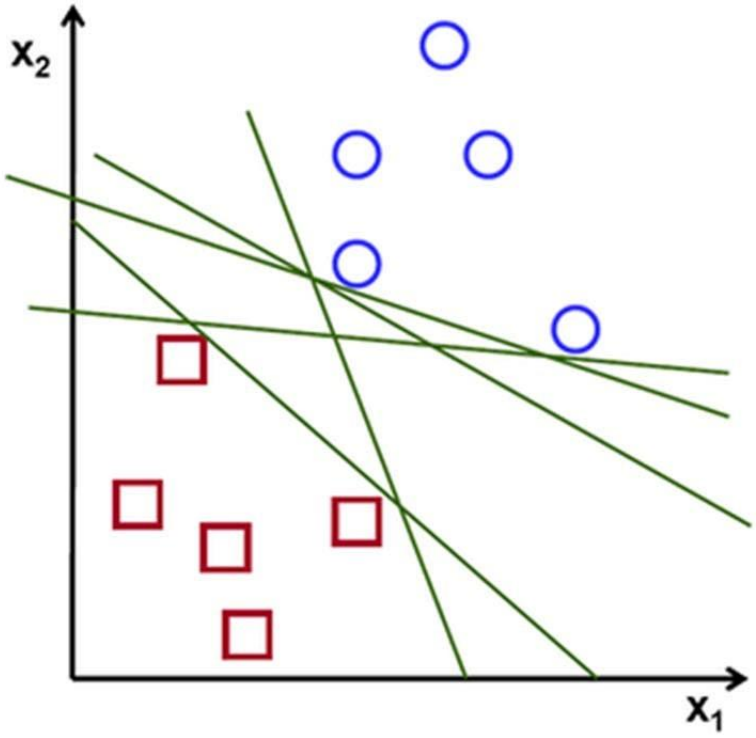


Figure 4.2. Green color hyperplane separating two classes of red squares and blue circles.

A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points.

Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM's theory. Therefore, the optimal separating hyperplane *maximizes* the margin of the training data which is depicted well in the Figure 7.2.

The optimal hyperplane can be represented in an infinite number of different ways by scaling of β and βo. As a matter of convention, among all the possible representations of the hyperplane, the one chosen is

$$|\beta_0 + \beta^T x| = 1 \quad (12)$$

Where x symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called support vectors. This representation is known as the canonical hyperplane.

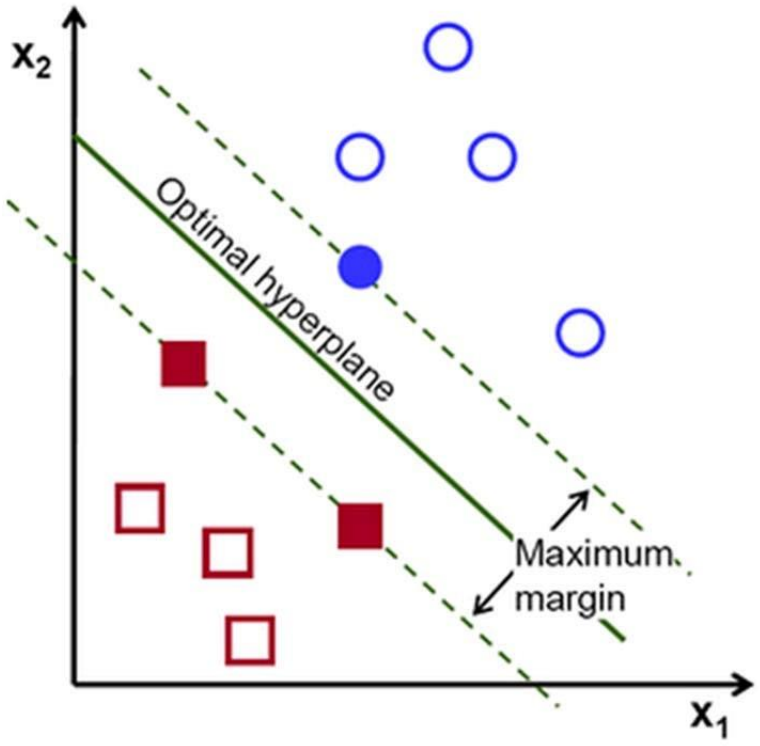


Figure 4.3. Finding an optimal hyperplane.

The data usually contain noises, which result in overlapping samples in pattern space, and there may produce some outliers in the training data set. So we need to remove these outliers from the training data set so that a better decision boundary can be easily formed. We here apply smoothening method to remove those points that do not agree with the majority of their k nearest neighbors.

In particular, by comparing with the 1-NN and k-NN classifiers, it can be found that the SVM classifier can not only save the storage space but also reduce the classification time under the case of no sacrificing the classification accuracy.

4.5 Histogram stretching to enhance the contrast:

Contrast is the difference between maximum and minimum pixel intensity. An important class of point operations is based upon the manipulation of an image histogram or a region histogram. The most important examples are described below.

Frequently, an image is scanned in such a way that the resulting brightness values do not make full use of the available dynamic range. This can be easily observed in the histogram of the brightness values shown in Figure 5.1. By stretching the histogram over the available dynamic range we attempt to correct this situation. If the image is intended to go from brightness 0 to brightness 2^B-1, then one generally maps the 0% value (or *minimum* as defined) to the value 0 and the 100% value (or *maximum*) to the value 2^B-1. The appropriate transformation is given by:

$$b[m,n] = (2^B - 1) \bullet \frac{a[m,n] - \text{minimum}}{\text{maximum} - \text{minimum}} \qquad (13)$$

This formula, however, can be somewhat sensitive to outliers and a less sensitive and more general version is given by:

$$b[m,n] = \begin{cases} 0 & a[m,n] \le p_{\text{low}}\% \\ (2^B - 1) \bullet \dfrac{a[m,n] - p_{\text{low}}\%}{p_{\text{high}}\% - p_{\text{low}}\%} & p_{\text{low}}\% < a[m,n] < p_{\text{high}}\% \\ (2^B - 1) & a[m,n] \ge p_{\text{high}}\% \end{cases} \qquad (14)$$

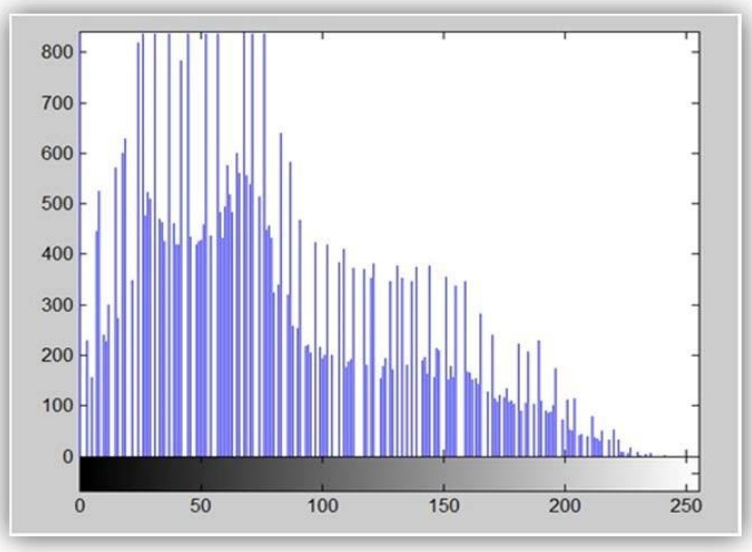


Figure 4.4. The stretched histogram of the image

In this second version one might choose the 1% and 99% values for $p_{low}\%$ and $p_{high}\%$, respectively, instead of the 0% and 100% values represented by eq. . . . It is also possible to apply the contrast-stretching operation on a regional basis using the histogram from a region to determine the appropriate limits for the algorithm. Note that in eqs. And it is possible to suppress the term 2^B-1 and simply normalize the brightness range to 0 <= $b[m,n]$ <= 1. This means representing the final pixel brightness's as reals instead of integers but modern computer speeds and RAM capacities make this quite feasible.

4.6 Contrast stretching:

Consider Figure 4.4. The histogram of the image is shown in Figure 4.5

Now we calculate contrast from this image.

Contrast = 225.

Now we will increase the contrast of the image. Increasing the contrast of the image: The formula for stretching the histogram of the image to increase the contrast is

$$g(x,y) = \frac{f(x,y)-fmin}{fmax- fmin} * 2^{bpp} \quad (15)$$

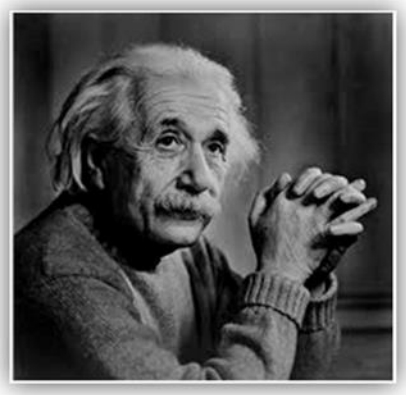


Figure 4.5. Image used for contrast stretching.

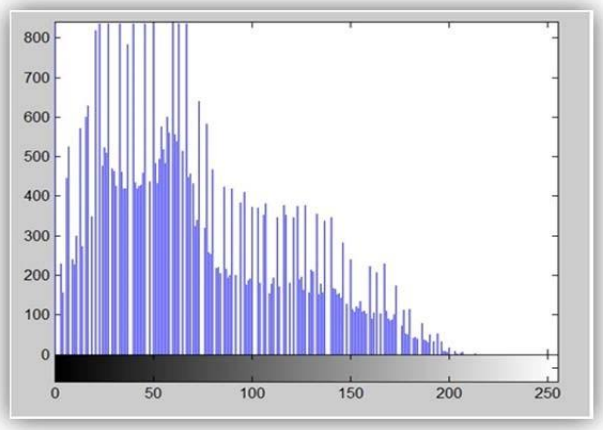


Figure 4.6 Histogram of the image.

The formula requires finding the minimum and maximum pixel intensity multiply by levels of gray. In our case the image is 8bpp, so levels of gray are 256. The minimum value is 0 and the maximum value is 225. So the formula in our case is

$$g(x,y) = \frac{f(x,y)-0}{225-0} * 255 \quad (15)$$

Where f(x,y) denotes the value of each pixel intensity. For each f(x,y) in an image, we will calculate this formula. After doing this, we will be able to enhance our contrast. The image in Figure 5 will appear after applying histogram stretching.

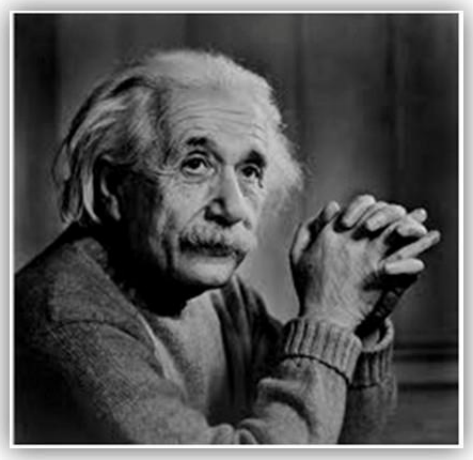


Figure 4.7. Image after applying histogram stretching.

The stretched histogram of this image has been shown in Figure 1. Note the shape and symmetry of histogram. The histogram is now stretched or in other means expands. Have a look at it.

In this case the contrast of the image can be calculated as Contrast = 240

Hence we can say that the contrast of the image is increased. Note: this method of increasing contrast does not work always, but it fails on some cases.

4.7 Sobel edge detection

Edge detection is more popular for identifying discontinuities in gray level than detecting isolated points and thin lines .The edge is the boundary between two regions with relatively distinct gray level properties. It is assumed here that the transition between two regions can be properties. The transition between two regions can be determined based on the gray level discontinuities .The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. In the input grayscale image, approximate gradient magnitude is also identified at each point by the edge detector. The operator

consists of a pair of 3x3 convolution kernels which is rotated by 90 degree [10]. The convolution masks of the Sobel detector are given in Figure 6.

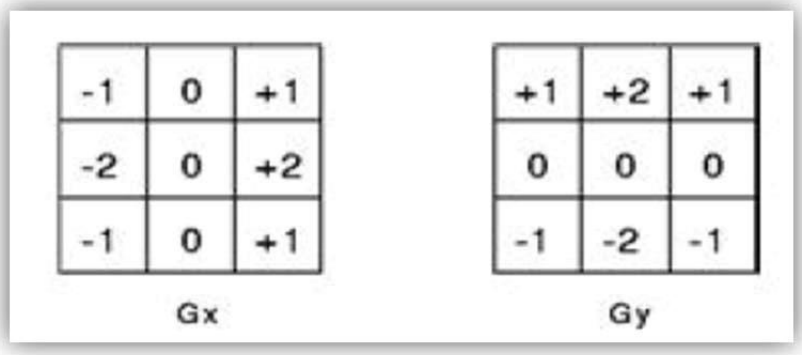


Figure 4.8. Sobel mask

Input: A Sample Image.

Output: Detected Edges.

Step 1: Accept the input image.

Step 2: Apply mask Gx,Gy to the input image.

Step 3: Apply Sobel edge detection algorithm and the gradient.

Step 4: Masks manipulation of Gx,Gy separately on the input image.

Step 5: Results combined to find the absolute magnitude of the gradient.

Step 6: The absolute magnitude is the output edges.

4.8 Proposed algorithm flow

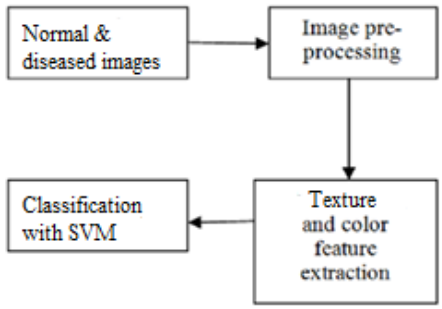
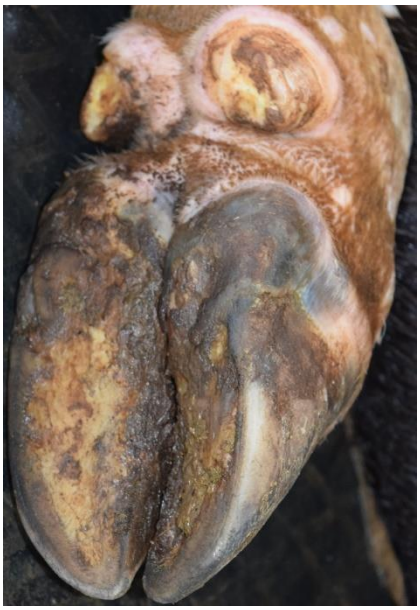
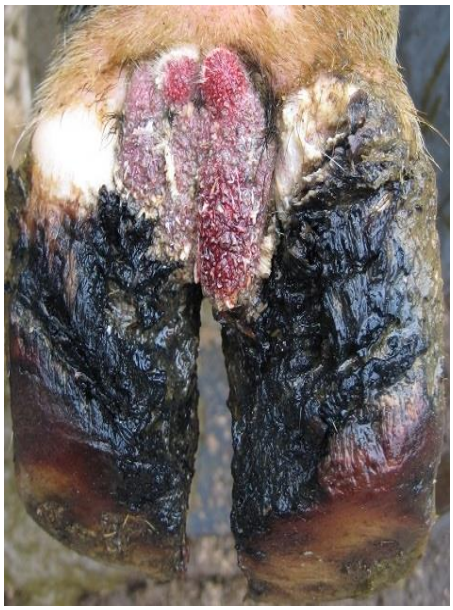


Figure 4.9 Algorithm flow

Main steps are data collection which contain normal and diseased images, image pre-processing, and feature extraction and classification

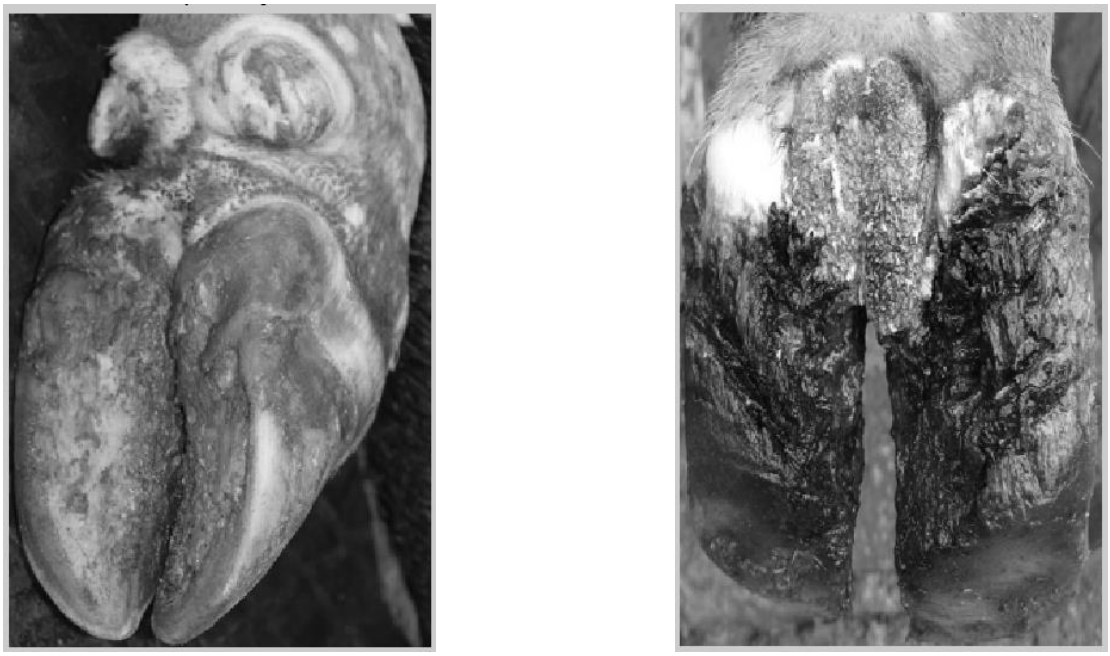
The image preprocessing stage consists of conversion of diseased and normal images to grayscale format and applying adaptive histogram to the grayscaled image. The image obtained after adaptive histogram equalization is considered for the extraction of various parameters.





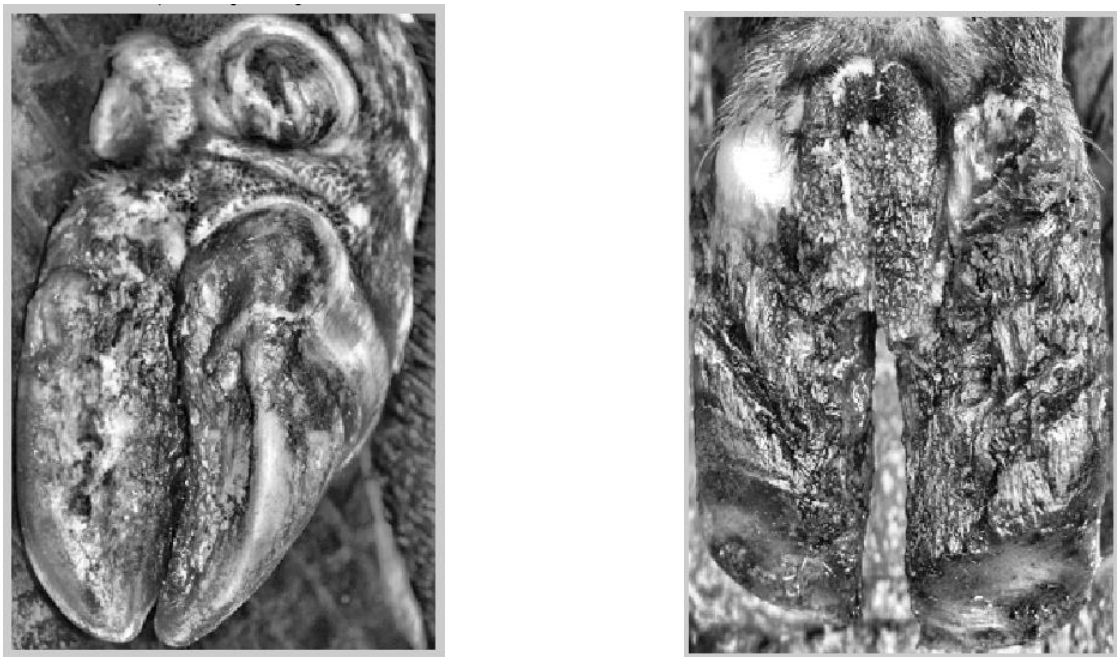
(a) **(b)**

Figure 4.10 (a) Normal foot image (b) Diseased foot image



(a) **(b)**

Figure 4.11 Gray scale image of (a) Normal foot (b) Diseased foot



(a) **(b)**

Figure 4.12 Adaptive histogram image of (a) Normal foot (b) Diseased foot

(a)

(b)

Figure 4.13 Results displayed for (a) Normal foot (b) Diseased foot

4.8.1 Calculated values of various parameters:

PARAMETER	NORMAL IMAGE	DISEASED IMAGE
Contrast	0.6683	1.3986
Correlation	0.8778	0.7681
Entropy	7.6014	7.7291
Energy	0.0697	0.0447
Homogeneity	0.7741	0.6834
Inverse difference movement	255	255
Kurtosis	2.1825	2.5981
Mean	102.0126	106.4884
RMS	15.9633	15.9475
Skewness	0.2909	0.5694
Smoothness	1.0000	1.0000
Standard deviation	55.5819	58.0504
Variance	2.2873e+03	3.1979e+03

CHAPTER 5

RESULTS AND DISCUSSION

The result of this project can be depicted as the comparison of the values of various parameters like Contrast, Correlation, Energy, Entropy, Homogeneity, Inverse difference movement (IDM), kurtosis, Mean, RMS, Skewness, Smoothness, Standard deviation, and Variance. Here we have considered a set of 50 images as a database, which will be considered as standard for comparison. Images considered as a standard database are shown below.

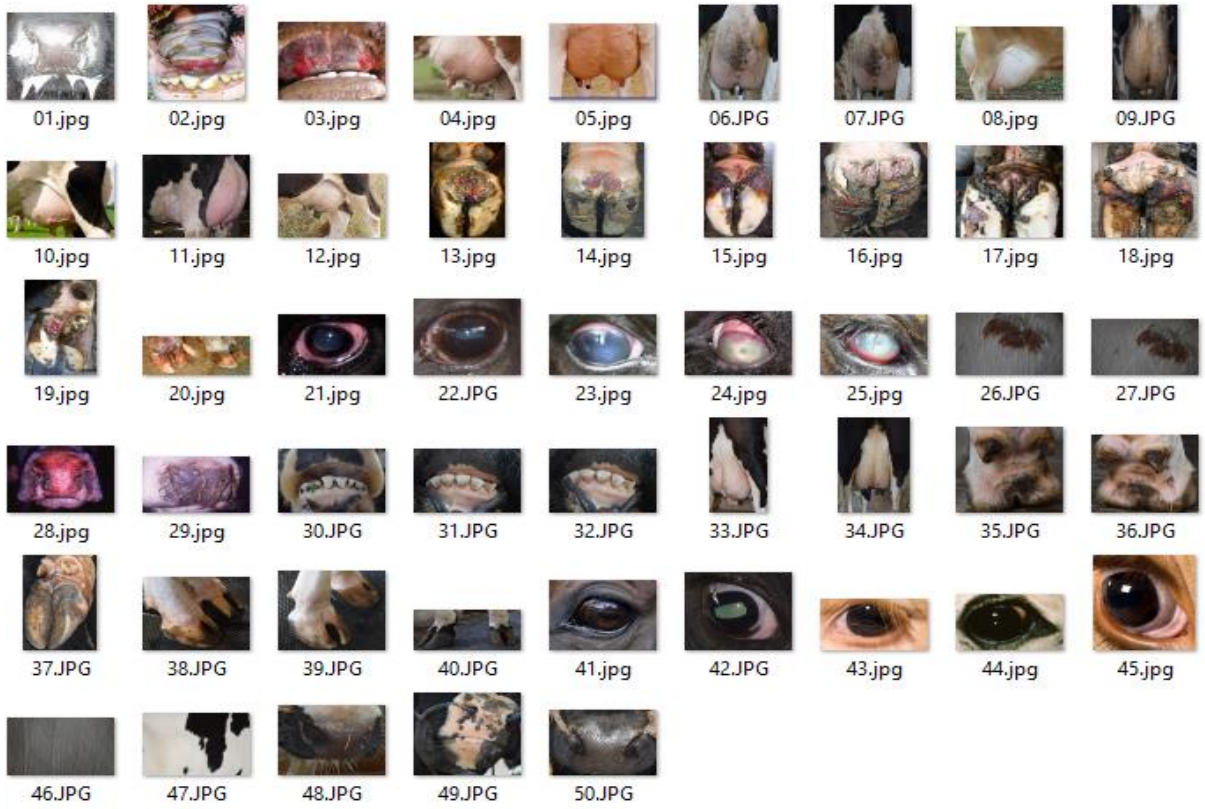


Figure 5.1 Images considered as database

The values obtained for all the images in database are saved in a file named dataset.mat Later, the images which need to be analyzed are stored in a test folder, from which images are processed one after the other and their obtained parameter values are compared with the standard

values from the dataset.mat file Depending on the closeness of the obtained and standard values, decision is made weather the cattle is diseased and if so, which disease, or it is decided as normal. An alerting message is sent to the concerned personnel if any animal is found to be diseased

Following are the parameter values obtained for the images shown in fig 5.1

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.7897	0.8785	0.0540	0.7429	162.9891	59.2780	7.5199	15.9677	2.4870e+03	1.0000	1.9724	0.0316	255
2	0.4726	0.9214	0.0759	0.8196	144.3904	67.1252	7.8703	15.9073	4.2092e+03	1.0000	2.1462	-0.4076	255
3	1.0996	0.8148	0.0503	0.7148	119.3163	55.2010	7.7550	15.9076	1.8897e+03	1.0000	2.5212	0.2315	255
4	0.4629	0.8957	0.0978	0.8343	132.0970	50.7879	7.6257	15.9646	1.8657e+03	1.0000	2.1775	0.0014	255
5	0.4894	0.8766	0.1099	0.8449	141.4435	50.4862	7.5327	15.9260	774.7317	1.0000	2.3898	-0.5032	255
6	0.7848	0.8481	0.0661	0.7629	108.6502	53.4889	7.5108	15.9443	2.5621e+03	1.0000	2.0793	-0.2184	255
7	0.6901	0.8620	0.0842	0.7871	73.3667	50.5494	7.1762	15.4903	2.2362e+03	1.0000	1.9726	0.3506	255
8	0.4506	0.8864	0.1141	0.8477	123.3370	47.4604	7.5226	15.9669	1.4908e+03	1.0000	2.1783	0.2473	255
9	0.8513	0.8142	0.0669	0.7391	76.8578	46.8044	7.3938	15.7619	1.2946e+03	1.0000	2.4119	0.4877	255
10	0.4270	0.9250	0.0872	0.8355	110.6352	80.3467	7.8102	14.9179	4.0675e+03	1.0000	1.7039	0.1589	255
11	0.4681	0.9001	0.0983	0.8287	84.6122	62.1781	7.2715	15.5610	2.5759e+03	1.0000	1.4888	0.2456	255
12	0.9771	0.8181	0.0647	0.7539	130.6314	58.6907	7.7233	15.9588	2.3167e+03	1.0000	1.8575	-0.1808	255
13	0.9771	0.8181	0.0647	0.7539	130.6314	58.6907	7.7233	15.9588	2.3167e+03	1.0000	1.8575	-0.1808	255
14	1.0190	0.8095	0.0572	0.7224	121.7715	56.6792	7.7236	15.9287	2.4079e+03	1.0000	2.0233	0.0122	255
15	1.0190	0.8095	0.0572	0.7224	121.7715	56.6792	7.7236	15.9287	2.4079e+03	1.0000	2.0233	0.0122	255
16	1.2467	0.7797	0.0528	0.7096	116.0696	71.0008	7.7888	15.9580	4.1986e+03	1.0000	1.9306	0.4711	255
17	0.8559	0.8601	0.0544	0.7431	112.5819	76.2393	7.9090	15.5226	4.8038e+03	1.0000	1.7984	0.3381	255
18	1.3848	0.7688	0.0443	0.6806	117.3209	61.7032	7.8950	15.8094	3.3573e+03	1.0000	2.1961	0.1965	255
19	0.6616	0.8886	0.0661	0.7885	113.4056	64.1968	7.8931	15.6148	3.6211e+03	1.0000	2.2541	0.3064	255
20	1.9535	0.6752	0.0358	0.6220	123.5628	52.2191	7.7174	15.9394	1.5526e+03	1.0000	2.4070	0.1496	255
21	0.2426	0.9443	0.1331	0.8886	38.2489	41.4585	6.6110	14.0440	1.3310e+03	1.0000	6.7844	1.9755	255
22	0.3869	0.9085	0.1109	0.8359	70.0110	31.4316	6.7939	15.9687	798.4262	1.0000	5.2682	1.2083	255
23	0.2950	0.9350	0.1052	0.8605	107.5703	56.7037	7.6388	15.9683	2.1013e+03	1.0000	2.7422	0.7704	255
24	0.2950	0.9350	0.1052	0.8605	107.5703	56.7037	7.6388	15.9683	2.1013e+03	1.0000	2.7422	0.7704	255
25	1.0867	0.8158	0.0520	0.7202	123.2571	53.3042	7.6928	15.9611	2.1006e+03	1.0000	2.5310	0.4286	255
26	1.5195	0.5571	0.0554	0.6301	87.8823	25.7953	6.5953	15.9485	556.6637	1.0000	3.1668	-0.8041	255
27	1.4600	0.5167	0.0613	0.6388	83.7411	24.4577	6.5293	15.9453	339.1473	1.0000	3.0945	-0.7675	255
28	0.5610	0.9042	0.1291	0.8132	68.7762	58.0074	7.0774	15.2697	1.6636e+03	1.0000	3.1308	0.9761	255
29	2.2271	0.6057	0.0391	0.6115	127.9302	67.0038	7.8993	15.9044	2.0372e+03	1.0000	1.8809	-0.0022	255
30	0.4842	0.9055	0.0835	0.8202	86.5015	43.9471	7.3678	15.8927	1.6807e+03	1.0000	2.0948	0.3173	255
31	0.8760	0.8140	0.0721	0.7609	74.5372	51.8075	7.3357	15.7774	1.3167e+03	1.0000	2.3863	0.7312	255
32	0.7713	0.8434	0.0748	0.7778	75.6767	53.5718	7.3340	15.7760	1.3826e+03	1.0000	2.3106	0.7215	255
33	0.4916	0.8942	0.0826	0.8073	95.4123	69.1623	7.4368	15.7717	3.6914e+03	1.0000	1.4839	0.3225	255
34	0.5063	0.8941	0.1095	0.8256	74.9191	56.5740	7.0542	15.8099	2.3691e+03	1.0000	1.8090	0.4939	255
35	0.7968	0.8196	0.0737	0.7600	82.8873	46.5752	7.4396	15.7687	1.3643e+03	1.0000	2.2610	0.4118	255
36	0.7400	0.8522	0.0672	0.7626	101.7658	55.3590	7.5973	15.9114	2.2765e+03	1.0000	1.7820	0.2483	255
37	0.6683	0.8778	0.0697	0.7741	102.0126	55.5819	7.6014	15.9633	2.2873e+03	1.0000	2.1825	0.2909	255
38	0.8988	0.8005	0.0659	0.7359	83.0800	63.2976	7.5601	15.6327	2.3567e+03	1.0000	2.3307	0.7598	255
39	0.6966	0.8473	0.0735	0.7660	89.9437	65.9353	7.4681	15.9461	2.1796e+03	1.0000	2.5843	0.8775	255
40	1.1879	0.7694	0.0544	0.7083	70.2083	37.8651	7.1625	15.8189	1.2904e+03	1.0000	3.2558	0.7123	255
41	0.3408	0.9271	0.0974	0.8461	77.9235	35.9128	7.0942	15.7504	1.2101e+03	1.0000	2.7829	-0.0960	255
42	0.3567	0.8952	0.1265	0.8437	48.5460	34.6621	6.2878	15.9631	980.8994	1.0000	5.6212	1.8384	255
43	0.3398	0.9258	0.0988	0.8516	129.9909	68.5917	7.9369	15.8550	2.2442e+03	1.0000	1.9077	-0.0741	255
44	0.6819	0.8784	0.1309	0.7957	80.8514	73.3774	7.3137	14.9831	3.1095e+03	1.0000	1.9324	0.6744	255
45	0.9997	0.8240	0.0574	0.7294	89.8613	67.5155	7.7812	15.0344	2.2087e+03	1.0000	2.1683	0.4906	255
46	1.2300	0.3837	0.0861	0.6514	91.1748	14.2043	5.8528	15.9687	106.0885	1.0000	2.4216	-0.1128	255
47	0.3424	0.9392	0.1436	0.8452	151.1359	82.5905	6.0989	15.7963	3.5062e+03	1.0000	1.9293	-0.9148	255
48	0.8472	0.7906	0.0748	0.7473	68.5815	40.8401	7.1005	15.8879	1.4934e+03	1.0000	3.1389	0.9795	255
49	0.4946	0.9015	0.0829	0.8085	91.1186	62.4956	7.5333	15.9359	2.4816e+03	1.0000	2.0535	0.6391	255
50	1.1518	0.7736	0.0509	0.6881	81.3202	43.1632	7.3186	15.8775	1.6664e+03	1.0000	3.7689	0.9510	255

5.1 Case study 1:

Consider a test image,

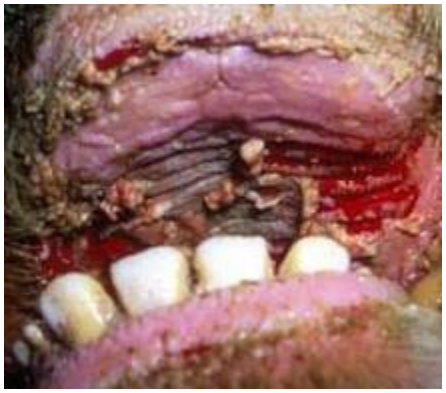


Figure 5.2 Image of Ulcer disease

PARAMETER	VALUES
Contrast	0.5537
Correlation	0.9007
Energy	0.0687
Homogeneity	0.7889
Mean	118.4271
Standard deviation	62.8167
Entropy	7.9070
RMS	15.7674
Variance	2.7008e+03
Smoothness	1.0000
Kurtosis	2.2114
Skewness	0.1700
Inverse difference movement	255

In the provided database first three images are given as the standard images for ulcer, and the test image considered gives highest matching with the values of those images, hence the result is displayed as ulcer disease

```
Command Window
result =

     1

disease type is ULCER
>>
```

Figure 5.3 Result shown for the test image considered in Case study 1

5.2 Case study 2:

Let us consider the image of a normal cattle

Figure 5.4 Image of Normal Eye

PARAMETER	VALUES
Contrast	1.2325
Correlation	0.7866
Energy	0.0529
Homogeneity	0.7110
Mean	99.3611
Standard deviation	62.4348
Entropy	7.7619
RMS	15.8987
Variance	2.8559e+03
Smoothness	1.0000
Kurtosis	2.4584
Skewness	0.5895
Inverse difference movement	255

The parameter values of the given test image is compared with all the values from database images, and it is found that the test image values match with the values of normal eye images. Hence the result is displayed as normal cattle

Figure 5.5 Result shown for the test image considered in Case study 2

The above case studies show the process flow for detection of the diseases and the procedure is same for all the remaining images

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

This project proposed a diseased cattle image pattern classification to identify disease in cattle with a combination of texture and color feature extraction. Initially the system sends a digital image of the diseased animals and these images are read in MATLAB and processed automatically based on SVM and the results are shown. The results of this project are to find appropriate features that can identify various diseases of cattle. Firstly, normal and diseased images are collected and pre-processed. Then, features of shape, color and texture are extracted from these images. After that, these images are classified by support vector machine classifier. A combination of several features are used to evaluate the appropriate features to find distinctive features for identification of cattle disease. When a single feature is used, shape feature has the lowest accuracy and texture feature has the highest accuracy. A combination of texture and color feature extraction results a highest classification accuracy. A combination of texture and color feature extraction with polynomial kernel results in good classification accuracy. Based on the classified type of disease a text message is sent to the user in the project.

6.2 Future scope:

In this project, we demonstrated only few types of diseases which were commonly caused and it can be extended for more disease in future. Here only a text message is sent to the veterinarian, automatically without human interaction.

This project can also be implemented by capturing the image using thermal (infrared) cameras. There we use temperature as a parameter for the detection of diseases. Its implementation is more costly compared to our now proposed project.

REFERENCES

[1] Sari, Yuita Arum, R V Hari Ginardi, Riyanarto Sarno. "Assessment of Color Levels in Leaf Color Chart UsingSmartphone Camera with Relative Calibration". Information Systems International Conference (ISICO), 2013: 631-636.

[2] Raid, Richard Neil, and J. C. Comstock. Sugarcane rust disease. University of Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, EDIS, 1998.

[3] Camargo, A., and J. S. Smith. "Image pattern classification for the identification of disease causing agents in plants." Computers and Electronics in Agriculture 66.2 (2009): 121-125.

[4] Asraf, H. Muhammad, M. T. Nooritawati, and M. S. B. Rizam. "A Comparative Study in Kernel-Based Support Vector Machine of Oil Palm Leaves Nutrient Disease." Procedia Engineering 41 (2012): 13531359.

[5] Phadikar, Santanu, Jaya Sil, and Asit Kumar Das. "Rice diseases classification using feature selection and rule generation techniques." Computers and Electronics in Agriculture 90 (2013): 76-85.

[6] Li, Daoliang, Wenzhu Yang, and Sile Wang. "Classification of foreign fibers in cotton lint using machine vision and multi-class support vector machine." Computers and electronics in agriculture 74.2 (2010): 274279.

[7] Ginardi, R. V Hari, RiyanartoSarno, and Tri AdhiWijaya. "Sugarcane Leaf Color Classification in Sa*b* Color Element Composition". 2013 International Conference on Computer, Control, Informatics and It's Application, pp:175-178.

[8] Li-jie, Yu, Li De-sheng, and Zhou Guan-ling. "Automatic Image Segmentation Base on Human Color Perceptions." International Journal of Image, Graphics and Signal Processing (IJIGSP) 1.1 (2009): 25.

[9] Shivakumar, G, P.A Vijaya. "Face Recognition Using Geometric Attributes". International Journal of Computational Intelligence Research Volume 6, Number 3 (2010), pp. 373–383.

[10] Shabanzade, Maliheh, MortezaZahedi, and Seyyed Amin Aghvami. "Combination of local descriptors and global features for leaf recognition." Signal and Image Processing: An International Journal (SIPIJ). v2 i3 (2011): 23-31.

[11] D. S. Huang. The local minima free condition of feedforward neural networks for outer-supervised learning. IEEE T. Syst. Man. Cy. B. 28:477–480, 1998.

[12] Y.-Y. Wan, J.-X. Du, D. S. Huang, Z. Chi, Y.-M. Cheung, X.-F. Wang, and G.-J. Zhang. Bark texture feature extraction based on statistical texture analysis. In Proceedings of the 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pages 482–485, Hong Kong, 2004.

APPENDIX A

A.1 MATLAB CODE FOR LOADING THE DATABASE IMAGES

```
clc
clear all
folder_name = uigetdir(pwd, 'Select the directory of images');

%dataset loading

pngImagesDir = fullfile(folder_name, '*.png');
jpgImagesDir = fullfile(folder_name, '*.jpg');
bmpImagesDir = fullfile(folder_name, '*.bmp');

% calculate total number of images
num_of_png_images = numel( dir(pngImagesDir) );
num_of_jpg_images = numel( dir(jpgImagesDir) );
num_of_bmp_images = numel( dir(bmpImagesDir) );
totalImages = num_of_png_images + num_of_jpg_images + num_of_bmp_images;

jpg_files = dir(jpgImagesDir);
png_files = dir(pngImagesDir);
bmp_files = dir(bmpImagesDir);

if ( ~isempty( jpg_files ) || ~isempty( png_files ) || ~isempty( bmp_files )
)
    % read jpg images from stored folder name
    % directory and construct the feature dataset
    jpg_counter = 0;
    png_counter = 0;
    bmp_counter = 0;
    for k = 1:totalImages

        if ( (num_of_jpg_images - jpg_counter) > 0)
            imgInfoJPG = imfinfo( fullfile( folder_name,
jpg_files(jpg_counter+1).name ) );
            if ( strcmp( lower(imgInfoJPG.Format), 'jpg') == 1 )
```

```
                % read images
                sprintf('%s \n', jpg_files(jpg_counter+1).name)
                % extract features
                image = imread( fullfile(folder_name,
jpg_files(jpg_counter+1).name ) );
                [pathstr, name, ext] = fileparts( fullfile(folder_name,
jpg_files(jpg_counter+1).name ) );
                image = imresize(image, [384 256]);
            end

            jpg_counter = jpg_counter + 1;

        elseif ( (num_of_png_images - png_counter) > 0)
            imgInfoPNG = imfinfo( fullfile( folder_name,
png_files(png_counter+1).name ) );
            if ( strcmp( lower(imgInfoPNG.Format), 'png') == 1 )
                % read images
                sprintf('%s \n', png_files(png_counter+1).name)
                % extract features
                image = imread( fullfile( folder_name,
png_files(png_counter+1).name ) );
                [pathstr, name, ext] = fileparts( fullfile( folder_name,
png_files(png_counter+1).name ) );
                image = imresize(image, [384 256]);
            end

            png_counter = png_counter + 1;

        elseif ( (num_of_bmp_images - bmp_counter) > 0)
            imgInfoBMP = imfinfo( fullfile( folder_name,
bmp_files(bmp_counter+1).name ) );
            if ( strcmp( lower(imgInfoBMP.Format), 'bmp') == 1 )
                % read images
                sprintf('%s \n', bmp_files(bmp_counter+1).name)
                % extract features
                image = imread( fullfile(folder_name,
bmp_files(bmp_counter+1).name ) );
```

```
                [pathstr, name, ext] = fileparts( fullfile(folder_name,
bmp_files(bmp_counter+1).name ) );
                image = imresize(image, [384 256]);
            end

            bmp_counter = bmp_counter + 1;

        end
     seg_img=image;
if size(seg_img,3) == 3
   img = rgb2gray(seg_img);
end

img = adapthisteq(img,'clipLimit',0.02,'Distribution','rayleigh');
% Create the Gray Level Cooccurance Matrices (GLCMs)
glcms = graycomatrix(img);

%Evaluate 13 features from the disease affected region only
% Derive Statistics from GLCM
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
n = size(seg_img,2);
```

```
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);

% Put the 13 features in an array
 feat_disease= [Contrast,Correlation,Energy,Homogeneity, Mean,
Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness,
IDM];
%set = [hsvHist autoCorrelogram color_moments texture vec1];
%set = [hsvHist autoCorrelogram color_moments];
dataset(k, :) = feat_disease ;
    end
end

diseasetype=[1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4 5 5 6 6 7 7 7
8 8 9 9 9 9 9 9 10 10 10 10 10 11 11 12 12 12];


%save ('dataset','dataset')
uisave({'dataset','diseasetype'},'dataset1')
```

A.2 MATLAB CODE FOR PROCESSING THE TEST IMAGE

```
close all
clear all
clc
%%
%  Select an image from the 'Disease Dataset' folder by opening the folder
% [filename,pathname] =
uigetfile({'*.*';'*.bmp';'*.tif';'*.gif';'*.png'},'Pick a Cattle Image');
% I = imread([pathname,filename]);
folder_name = uigetdir(pwd, 'Select the directory of test images');

jpgImagesDir = fullfile(folder_name, '*.jpg');
num_of_jpg_images = numel( dir(jpgImagesDir) );
jpg_files = dir(jpgImagesDir);
jpg_counter = 0;

for k = 1:num_of_jpg_images

    pause(5)
    close all
%%
I = imread( fullfile(folder_name, jpg_files(jpg_counter+1).name ) );
figure, imshow(I);title('Cattle Image');
jpg_counter=jpg_counter+1;

 image = imresize(I, [384 256]);

      seg_img=image;
if size(seg_img,3) == 3
   img = rgb2gray(seg_img);
end
figure, imshow(img);title('Grayscale Image');
img = adapthisteq(img,'clipLimit',0.02,'Distribution','rayleigh');
figure, imshow(img);title('Adaptive Histogram Image');
%%
% Create the Gray Level Cooccurance Matrices (GLCMs)
```

```
glcms = graycomatrix(img);

%Evaluate 13 features from the disease affected region only
% Derive Statistics from GLCM
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
n = size(seg_img,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);

% Put the 13 features in an array
 feature= [Contrast,Correlation,Energy,Homogeneity, Mean, Standard_Deviation,
Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];

 load dataset1.mat
% 'diasesefeat' contains the features of the disease affected leaves of both
```

```
% the types
% 'diseasetype' contains the corresponding label
% Train the classifier
%svmStruct = svmtrain(dataset,diseasetype);
% Classify the test image
%species_name = svmclassify(svmStruct,feature)
result = multisvm(dataset,diseasetype,feature)

mbn='8710969151';

if k < 5
    c=1;
elseif k > 5 && k < 11
    c=2;
elseif k > 10 && k < 16
    c=3;
elseif k > 15 && k < 21
    c=4;
elseif k > 20 && k < 26
    c=5;
 elseif k > 25 && k < 31
    c=6;
 elseif k > 30 && k < 36
    c=7;
end
switch result
    case 1
        disp('disease type is ULCER')
      %  msg='disease type is ULCER';
     %   fmsg=strcat(msg,'cattle No:',num2str(c));
    %gsmsend(mbn,fmsg);
       case 2
        disp('disease type is udder inflammation')
     %   msg='disease type is udder inflammation';
      %  fmsg=strcat(msg,'cattle No:',num2str(c));
    %gsmsend(mbn,fmsg);
       case 3
        disp('Foot Disease')
```

```
      %  msg='Foot Disease';
     %   fmsg=strcat(msg,'cattle No:',num2str(c));
    %gsmsend(mbn,fmsg);
    case 4
        disp('PINK EYE disease')
        %msg='EYE disease';
       % fmsg=strcat(msg,'cattle No:',num2str(c));
%    gsmsend(mbn,fmsg);
    case 5
        disp('Skin Disease')
       % msg='Skin Disease';
        %fmsg=strcat(msg,'cattle No:',num2str(c));
    %gsmsend(mbn,fmsg);
    case 6
        disp('Nasal disease')
        % msg='Nasal Disease';
        %fmsg=strcat(msg,'cattle No:',num2str(c));
    %gsmsend(mbn,fmsg);
    case 7
        disp('Normal Cattle')
    case 8
        disp('Normal Cattle')
    case 9
        disp('Normal Cattle')
    case 10
        disp('Normal Cattle')
    case 11
        disp('Normal Cattle')
    case 12
        disp('Normal Cattle')

end
end
```

A.3 MATLAB CODE FOR REALTIME APPLICATION

```
clc;
clear all;
close all;



vid=videoinput('winvideo',1,'YUY2_320X240');

set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 3;

start(vid)
preview(vid)
% disp('press any key to continue..')
% pause
for k=1:20
 pause(2)
        data = getsnapshot(vid);
    figure(1)
    imshow(data)
    title('captured image')
    I=data;
 image = imresize(I, [384 256]);


      seg_img=image;
if size(seg_img,3) == 3
   img = rgb2gray(seg_img);
end
figure(2), imshow(img);title('Grayscale Image');
img = adapthisteq(img,'clipLimit',0.02,'Distribution','rayleigh');
figure(3), imshow(img);title('Adaptive Histogram Image');
%%
% Create the Gray Level Cooccurance Matrices (GLCMs)
glcms = graycomatrix(img);
```

```
%Evaluate 13 features from the disease affected region only
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
n = size(seg_img,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);

% Put the 13 features in an array
 feature= [Contrast,Correlation,Energy,Homogeneity, Mean, Standard_Deviation,
Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];

 load dataset1.mat
% 'diseasetype' contains the corresponding label
% Train the classifier
%svmStruct = svmtrain(dataset,diseasetype);
% Classify the test image
%species_name = svmclassify(svmStruct,feature)
```

```
result = multisvm(dataset,diseasetype,feature)


mbn='8710969151';


if k < 5
    c=1;
elseif k > 5 && k < 11
    c=2;
elseif k > 10 && k < 16
    c=3;
elseif k > 15 && k < 21
    c=4;
elseif k > 20 && k < 26
    c=5;
 elseif k > 25 && k < 31
    c=6;
 elseif k > 30 && k < 36
    c=7;
end


switch result
    case 1
        disp('disease type is ULCER')
       %imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
       %msg='disease type is ULCER';
       %fmsg=strcat(msg,'cattle No:',num2str(c));
       %gsmsend(mbn,fmsg);


       case 2
        disp('disease type is udder inflammation')
        %imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
        %msg='disease type is udder inflammation';
        %fmsg=strcat(msg,'cattle No:',num2str(c));
        %gsmsend(mbn,fmsg);
       case 3
        disp('Foot Disease')
       % imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
```

```
        % msg='Foot Disease';
        % fmsg=strcat(msg,'cattle No:',num2str(c));
        % gsmsend(mbn,fmsg);
    case 4
         disp('EYE disease')
        % imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
        % msg='EYE disease';
        % fmsg=strcat(msg,'cattle No:',num2str(c));
        % gsmsend(mbn,fmsg);
    case 5
         disp('Skin Disease')
        % imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
        % msg='Skin Disease';
        % fmsg=strcat(msg,'cattle No:',num2str(c));
        % gsmsend(mbn,fmsg);
    case 6
         disp('Nasal Cattle')
        % imwrite(data,strcat(num2str(k),'_',num2str(result),'.jpg'))
        % msg='Nasal Disease';
        % fmsg=strcat(msg,'cattle No:',num2str(c));
        % gsmsend(mbn,fmsg);
    case 7
         disp('Normal Cattle')
    case 8
         disp('Normal Cattle')
    case 9
         disp('Normal Cattle')
    case 10
         disp('Normal Cattle')
    case 11
         disp('Normal Cattle')
    case 12
         disp('Normal Cattle')

end
end
closepreview(vid);
stop(vid);
```

A.4 MATLAB CODE FOR CLASSIFICATION OF TEST IMAGE USING MULTISVM

```
function [itrfin] = multisvm( T,C,test )
%Inputs: T=Training Matrix, C=Group, test=Testing matrix
%Outputs: itrfin=Resultant class
%C=str2num(cell2mat(C));
itrind=size(test,1);
itrfin=[];
Cb=C;
Tb=T;
for tempind=1:itrind
    tst=test(tempind,:);
    C=Cb;
    T=Tb;
    u=unique(C);
    N=length(u);
    c4=[];
    c3=[];
    j=1;
    k=1;
    if(N>2)
        itr=1;
        classes=0;
        cond=max(C)-min(C);
        while((classes~=1)&&(itr<=length(u))&& size(C,2)>1 && cond>0)
        %This while loop is the multiclass SVM Trick
            c1=(C==u(itr));
            newClass=c1;
            %svmStruct = svmtrain(T,newClass,'kernel_function','rbf'); % I am
using rbf kernel function, you must change it also
            svmStruct = svmtrain(T,newClass);
            classes = svmclassify(svmStruct,tst);

             if classes~=1
              %  itr=itr+1;

            % This is the loop for Reduction of Training Set
```

```
            for i=1:size(newClass,2)
                if newClass(1,i)==0;
                    c3(k,:)=T(i,:);
                    k=k+1;
                end
            end
        T=c3;
        c3=[];
        k=1;
            % This is the loop for reduction of group
            for i=1:size(newClass,2)
                if newClass(1,i)==0;
                    c4(1,j)=C(1,i);
                    j=j+1;
                end
            end
        C=c4;
        c4=[];
        j=1;

        cond=max(C)-min(C); % Condition for avoiding group to contain similar
type of values and the reduce them to process
% This condition can select the particular value of iteration base on classes
            if classes~=1
                itr=itr+1;
            end
             end
        end
    end

valt=Cb==u(itr);        % This logic is used to allow classification
val=Cb(valt==1);        % of multiple rows testing matrix
val=unique(val);
itrfin(tempind,:)=val;
end

end
```

A.5 MATLAB CODE FOR SENDING MESSAGE VIA GSM MODULE

```
% word= 'AT'
function gsmsend(mobno,msg)
s=serial('COM13');%initialize com port as serial & set baud rate for laptop
check with
  set(s,'BaudRate',9600);

% open the com port and start sending
   fopen(s);
   fprintf(s,'%s','AT')
   %fprintf(s,'%s','D')
   fprintf(s, '%c', 13)
   pause(2)
   fprintf(s,'%s','AT+CMGF=1')
   %fprintf(s,'%s','D')
   fprintf(s, '%c', 13)
   pause(2)
     %fprintf(s,'%s','AT+CMGS="9448847874"')
    fprintf(s,'%s', strcat('AT+CMGS="',mobno,'"'));
   %fprintf(s,'%s','D')
   fprintf(s, '%c', 13)
   pause(2)
     fprintf(s,'%s',msg)
%      fn=12,PSNR=24,SSIM=0.25;
%     fprintf(s,'async','%s: PSNR = %3.2f  SSIM = %f \n',PSNR, SSIM)
   %fprintf(s,'%s','D')
   fprintf(s, '%c', 26)
   pause(2)
 % out = fscanf(s,'%s')
   fclose(s);

%     this.serialPort.WriteLine("AT" + (char)(13));
%                     Thread.Sleep(2000);
%                     this.serialPort.WriteLine("AT+CMGF=1" + (char)(13));
%                     Thread.Sleep(3000);
```

```
%                       this.serialPort.WriteLine("AT+CMGS=\"" + cellNo +
"\"");
%                       Thread.Sleep(5000);
%                       this.serialPort.WriteLine(">" + messages + (char)(26));
```