Identification and Classification of Rice Varieties using Image Processing & Soft Computing Techniques

Prepared By

Swarna Kamal Dhyawala
Registration No. 151010110054 of 2015-2019

Ankit Srivastava
Registration No. 151010110009 of 2015-2019

Souvik Pal
Registration No. 161010120014 of 2015-2019

Pema Lamu Bhutia
Registration No. 161010120010 of 2015-2019

Under the guidance of

Prof. Dhiman Mondal
Head of Department
Department of Computer Science & Engineering
Jalpaiguri Government Engineering College

PROJECT REPORT

Submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

Department of Computer Science & Engineering
Jalpaiguri Government Engineering College

Affiliated to

Maulana Abul Kalam Azad University of Technology, West Bengal
May 2019

CERTIFICATION

Jalpaiguri Government Engineering College

Faculty of Computer Science & Engineering

This is to certify that the work in preparing the project entitled “*Identification and Classification of Rice Varieties using Image Processing Techniques”* has been carried out by Swarna Kamal Dhyawala, Ankit Srivastava, Souvik Pal and Pema Lamu Bhutia under my guidance during the session 2015-2019 for the degree of Bachelor of Technology in Computer Science & Engineering.

…………………………..........
Prof. Dhiman Mondal
Head of Department
Computer Science & Engineering
Jalpaiguri Government
Engineering College

…………………………..........
Prof. Dhiman Mondal
Supervisor
Computer Science & Engineering
Jalpaiguri Government
Engineering College

Preface

In this project, we present an automatic evaluation method for the determination of the variety of rice. Among the rice samples, the variety of rice are determined with the help of shape descriptors and geometric features. Each rice variety of the nine rice varieties were represented by corresponding labels of size, shape and varietal types. This proposed method gives good results in evaluation of rice variety.

We would like to extend our gratitude to Prof. Dhiman Mondal from the Department of Computer Science & Engineering for his contributions towards the image processing and machine learning components of our system. Furthermore, this project and documentation would never have been more educative and efficient without his constant help and guidance. We would like to thank her for his guidance and for always being a source of encouragement.

We would like to thank the **Bidhan Chandra Krishi Viswavidyalaya** for consent to collect images of rice grains as a part of our project. We would like to thank **Dr.Kushal Rai** , who helped us in image acquition process.

We also express our deepest and sincere gratitude to all our teachers especially **Dr. Dipak Kumar Kole** , **Prof. Chinmoy Ghosh** and **Prof. Srinibas Rana** for their kind comments and advice during the different stages of our project. We are also extremely grateful to **Prof. Subhas Barman**, **Prof. Animesh Hazra, Prof. Jhuma Dutta** and **Dr. Amitava Ray**(Principal) for their encouragement and support. Last but not least, we would like to thank the librarians of our institution for their constant support.

…………………………… ……………………………

Swarna Kamal Dhyawala Ankit Srivastava

…………………………… …………………………

Souvik Pal Pema Lamu Bhutia

Index

Chapter 1

Introduction………………...……… 1
1.1 Terminologies and Definitions…………………..……………...………. 1
1.2 Problem Definition……………………………..…………………….. 1
1.3 Objective of the Project………………………………..……………….. 1-2
1.4 Tools and Platforms..………………………………………..……………3
1.4.1 MATLAB………………………………………..…………. 3
1.4.2 R………………………………………..…………….. 3

Chapter 2

Literature Survey……………………………………………………………….. 4-5

Chapter 3

Architecture of System 6-7

Chapter 4

Preprocessing…………………..……………………………………………………… 8-11
4.1 Image Acquisition………………………………………………………. 8
4.2 Image Preprocessing...…………. 8
4.2.1 Grey Scale Conversion…..………………………………… 8
4.2.2 Labelling of Region……………………..………………… …9
4.2.3 Image Segmentation……………………………………… 9
4.2.4 Background Substraction………………………………… 10
4.2.5 Image Orientation…………………………………………. 11

Chapter 5

Feature Extraction……..……………………………………………………….. 12-15
5.1 System Phases…………………………..………………………………. 12-15
5.1.1 Morphological Feature………..…………………………….12-14
5.1.2 Texture Feature..…………………………………………..... 14-15
5.2 Dataset Creation………..……………………………………………….. 15

Chapter 6

Feature Selection……………………………………………………………... 16

Chapter 7

Classification……….……………………………………………………………… 17-32

7.1 Random Forest……………………………………………. 17

7.2 Neural Network(ANN)…………………………………… 22

7.3 QDA………………………………………………………. 28

7.4 MARS………………………………… ……………… 30

.

Chapter 8

Result & Discussion………………………………………………………………33-38

8.1 Random Forest……………………………………………. 33

8.2 Neural Network(ANN)…………………………………… 34

8.3 QDA………………………………………………………. 35

8.4 MARS………………………………… ……………… 36

List of Figures

1.1 Various Variety of Rice Species…………………………..........…...…………... 2

2.1 Comparison Table…………………………………..…………………………... 5

3.1 Architecture of System……………………………………...…………………... 6

3.2 Different Phases of Proposed System…...…………………………..……………. 7

4.1 Red Plane…………………………………………….……………….…………. 9

4.2 Green Plane ……..……………………………..……………….…………….. 9

4.3 Blue Plane……………………..…………………………….…………………… 9

4.4 Binary image with noise……..……………………………………………….….. 9

4.5 Binary image without noise ….………………………………………….…… 9

4.6 Background Subtracted Image ………..…….. ………………………………..10

5.1 Height & Width ……………………..…………………..…………….……… 12

5.2 Dataset of Danti…………………... ……………………………………..… 15

5.3 Dataset of Harnana……………………………………………………..…....... 15

5.4 Dataset of Sabraj…………………………………………………………....… 15

7.1 A Plot of Classifier under Random Forest……………………………….....… 18

7.2 A decision tree ……………..……………………………………………… 19

7.3 Random forest as an ensemble……………………………………………….. 20

7.4 Working of a random forest classifier…………………………………………… 21

7.5 Training and testing phase of Random forest classifier……………………….... 21

7.6 An Artificial neural network with three layers…………………………...….. 22

7.7 A single neuron with its inputs and activation function…………………………. 23

7.8 Plot of the Heaviside function………………………………………….…….. 24

7.9 Plot of the sign function…………………………………………..……….. 25

7.10 Plot of the linear function……………………………….……….………… 25

7.11 Plot of the sigmoid function……………………………………………....... 26

7.12 Plot of the hyperbolic tangent function…………………………….………………….. 26

7.13 The backpropagation mechanism…………………………….………………….. 27

7.14 Comparison Between LDA and QDA……………………...………………….. 29

7.15 MARS Model……………………………………………...………………… 31

8.1 Random Forest - 10 Cross Validation………………………………….……... 33

8.2 Random Forest - Confusion Matrix………………………………………….33

8.3 Random Forest - Accuracy………………………………………………….. 34

8.4 Neural Network - 10 Cross Validation……….…………………………….. 34

8.5 Neural Network - Confusion Matrix…………………………………..…… 34

8.6 Neural Network - Accuracy…………………………………………………… 35

8.7 QDA - 10 Cross Validation……….…………………………………………… 35

8.8 QDA - Confusion Matrix……………………………..…………………….. 35

8.9 QDA - Accuracy…………………………………………...…………….. 36

8.10 10 Best Feature Selected using MARS………….., …………………………….. 36

8.11 MARS - Confusion Matrix………………………………..……………...... 36

8.12 MARS - Accuracy……………………………….…………………….. 37

8.13 Performance Data for the Neural Network & Random Forest Classifier……………………………………………………………………. 37

8.14 Comparative Study of Proposed Method with Prevailing Method…………………………………………....……………….... 38

Chapter 1

Introduction

1.1 Terminologies and Definitions:

Rice is the seed of the grass species **Oryza** sativa (Asian **rice**) or **Oryza** glaberrima (African **rice**). As a cereal grain, it is the most widely consumed staple food for a large part of the world's human population, especially in Asia.

Feature extraction is the process of extracting details about a particular object or of an image as image as whole which helps in uniquely identifying it.

Classification is the process of allotting a class to an object or an observation based upon the conventional features of the class is called classification.

Identification is a psychological process whereby the individual assimilates an aspect, property, or attribute of the other and is transformed wholly or partially by the model that provides.

Image Processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it.

1.2 Problem Definition

Our aim is to classify the rice grains based on the feature which we will extract. The intension of this work is to develop a real time application which was used to identify and classify of the rice grains.

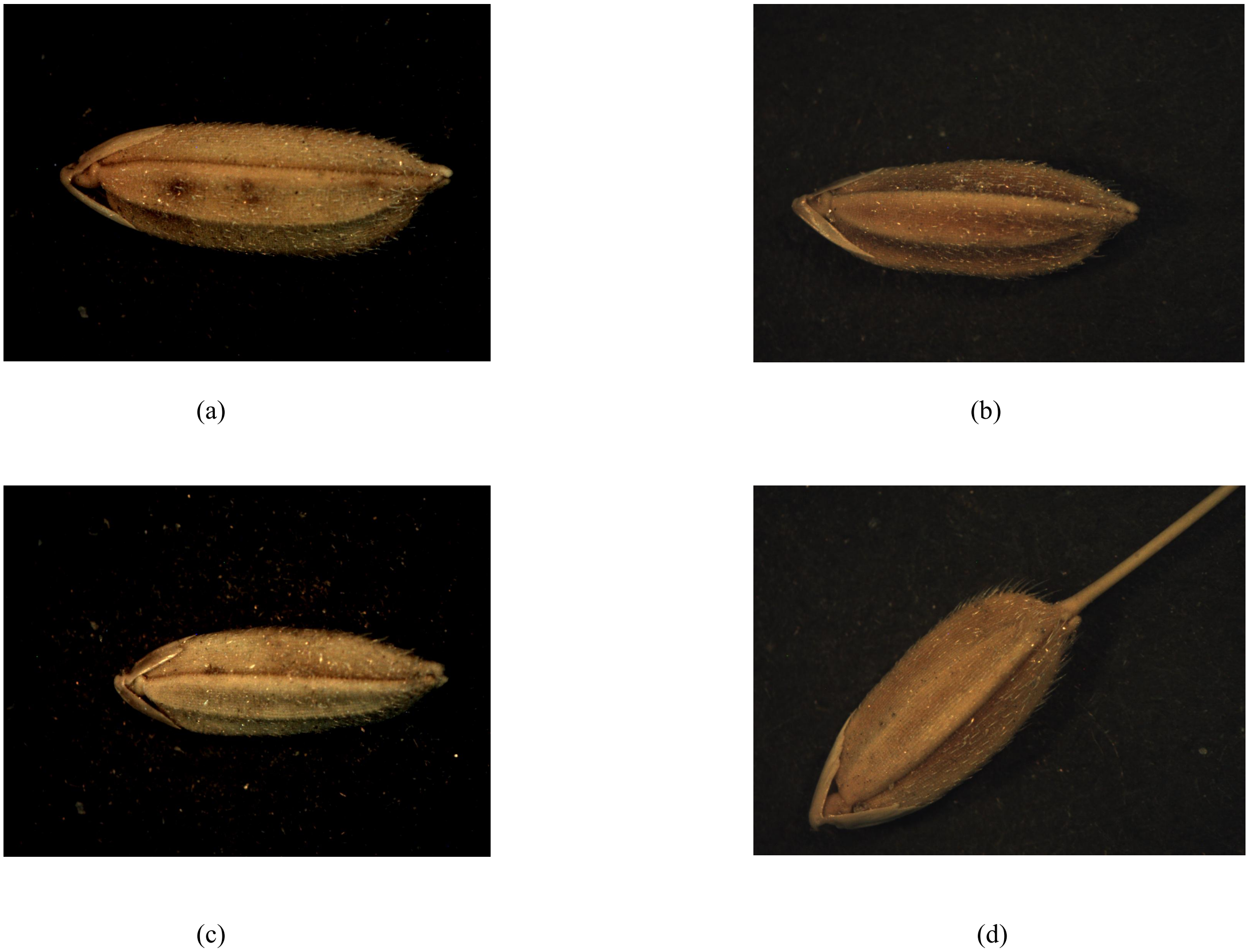
1.3 Objective of the Project

Rice, Scientifically known as ‘Oryza sativa’, is one of the most grained and consumed food crop. As the variety of rice is huge so is the demand for its identification and classification of species. Classification of rice grains is a challenge since manual classification that is being used in this industry which may not be objective or efficient. classification of rice is mainly defined by its chemical & physical characteristics. In the present grain classification system, grain category rapidly assessed by visual inspection. However, this process is annoying and time-consuming. so rather than manual process of classification, an automated system is introduced where digital images of rice species are recognized to be extracted the features efficient and faster way.

The demand for quality of food products we consume is increasing day by day. As the literacy rate is increasing in India so is the need for quality of food products is increasing. India is the second largest producer of rice grains first being China. As the production of rice is increasing so is the demand for its quality. This demand for quality of food grains is increasing because some of the traders cheat the shopkeepers by selling poor quality food grains which contains foreign particles like stones, sand, leaf, broken and damaged seeds etc.

As the technology is growing wider people are adopting the new technologies rather than using the old techniques. The growth in technology is making people more demanding towards the things they use and consume, this is the reason why everything is becoming automated. The use of Image processing techniques for testing the quality of rice grains is inexpensive and is less time consuming. The quality of grain is tested based on its color, size, shape and texture features in this method.

This project report provides the technique by which the features of the rice species are extracted with accuracy. Here different varieties of rice grains like danti, pakistani basmati, radhuni pagol and other more hundred varieties are taken. The morphological features such as height, width, area, perimeter, Major & Minor Axis Length, Eccentricity, angle, aspect ratio etc. are extracted from the digital image of rice. An algorithm for identification and classification of different varieties of rice, using the color and morphological features is presented. The proposed algorithm consists of several steps: Image acquisition, Image Segmentation, Feature selection and Extraction, identification and classification.



(a) (b) (c) (d)

Fig. 1.1 *Various Variety of Rice Species*
(a) Harnana (b) Kamal (c) Danti (d) Sabraj

1.4 Tools and Platforms

1.4.1 MATLAB

Matlab is a fourth generation programming language that is developed by MathWorks, an American private, mathematical software development company. Matlab helps in implementing algorithms, in plotting data, in developing user interface and also in simplifying matrix operations.Interfacing other programs written in C, C++, Python, etc is possible on Matlab. It is regarded as the most common language for Image Processing, but its use is limited to the authentic customers and developers who have been granted the trial version for learning purpose. Matlab requires a good processor speed and RAM for running successfully. The Matlab scripts are stored as files with a “.m” extension.

In this project, the most frequently used functions are available in the Image Processing toolbox . This toolbox provides a variety of functions with different prototypes to enable the user to perform different mathematical and graphical operations on the images. Some of its methods are specific to images while some others are common to all matrices. It is possible to perform image enhancement, noise reduction, object recognition, feature extraction, contrast adjustment, etc on images. Some of the functions are multithreaded to make use of a multiprocessor environment. Some commonly used algorithms in Image Processing have been developed and stored as predefined functions which gives Matlab an edge over other programming languages. Canny’s edge detector, Sobel’s edge detector, etc can be used to find the region of interest in images and we have used them to detect and locate the optic disc and also to study its features.

1.4.2 R

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, statistical inference to name a few. Most of the R libraries are written in R, but for heavy computational task, C, C++ and Fortran codes are preferred.

The primary uses of R is and will always be, statistic, visualization, and machine learning.R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. Many of R's standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made.

Chapter 2

Literature Survey

In this context number of author proposed their work and is what follows.

Megha and Kulkarni, [1] have proposed Classification and Quality Analysis of rice Grains. The intension of this work was to develop a real time application which was used to identify and classify of the rice grains and to grade (grade1, grade2, grade3) grains. The grading of rice sample is done according to the presence of impurities and size of the grain. The image samples were captured using camera and the images were stored in a database. The geometric feature and color features were considered in their work.

The color feature includes red color, green color, blue color whereas Geometric features include the length, width, and the shape of the rice grain. Chetna V. Maheshwari [2] proposed image processing techniques for identifying two varieties of rice based on shape and size. Image of a sample spread on the black paper were captured using a camera, the edge detection operation were performed to calculate the different parameters. Based on these parameters they classified rice seeds into three parts namely normal, small and long rice seeds and displayed the count of normal, small and long rice seeds on screen. Harpreet Kaur & Baljit Singh [3] they proposed a technique for classification of rice grains using multi-class SVM (support vector machine). He collected sample image by spreading the grains on glass of scanner and using black sheet of paper as background. Firstly the images were preprocessed to remove noise. Then different techniques are used such as Smoothening, Segmentation, Binarization. The content of chalky grains were calculated, further the chalky degree is calculated. The length of the rice grain is analyzed by using grain shape and it is depends on its length and width of the rice grain. If its length is greater than 75% than it's considered as unbroken of else it's taken as broken. Here an SVM (support vector machine) which is a new type of classifier, where the grains were classified as grade A , grade B, grade C. Bhavesh B. Prajapati, Sachin Patel [4] proposed algorithm for quality analysis of Basmati Rice using image processing techniques. They say with the help of this algorithm, an automated software system can be made to avoid the human inspection and related drawbacks. Image processing techniques can classify the rice grain with speed and accuracy. Photographic enlarger is used to measure the dimensions and to obtain the average length and width ratio of the basmati grain. G.Ajay, M.Suneel, K.Kiran Kumar, P.Siva Prasad [5] are proposed a quality evaluation of rice grains using morphological methods. Grains were said to be broken kernels whose lengths were 75% of the rice grain size. Features like length, width, and perimeter are considered. Morphological operations like erosion and dilations are carried out. The image processing command is used to convert an image to gray scale image. Firstly the morphological features are extracted and the length of the rice grains are obtained and to set a threshold value for the length of grains. The grains whose values are less than the threshold (TH) were considered as broken grains. Whereas those grains are greater than the threshold (TH) value were considered as whole grains. The method is computationally efficient to perform the rice classification whether broken or not by the method is faster and simple. Jagdeep Singh Aulakh , Dr. V.K. Banga [6] proposed the techniques for classification of rice grains by using image processing technique. The classification of grain is done according to the size of the grain (full, half or broken). The grains images were captured by using a Flat Bed Scanner (FBS), and also by using high resolution camera. Image thus acquired is then converted to binary image. And then apply morphological operations. And find out the properties of connected components of binary image. The object feature were extracted like Connectivity, Image size, Num objects, of image and based on these feature graph were plotted and the grain kernels which have lesser values than a threshold were discarded. And lastly calculate the number of full length rice grains in the sample image to grade the quality.

Substantial work for classifying and grading of rice grains has been reported. Neelamegam et al., [7]described a method for gradation and classification of different rice grains.An artificial neural network approach is used in the Identification and classification of the rice grain samples. Sukhvir Kaur et al., [8] presented a work on Geometric Feature Extraction of Selected Rice Grains using Image Processing Techniques. Analysis of Rice Granules using Image Processing and Neural Network Pattern Recognition Tool is been implemented by Abirami et al [9]. Priyankaran Tanck, Bipan Kaushal [10] presented the paper proposes a digital method which can be used to evaluate the quality of rice for the present Agmark Standards formulated with the help of digital image processing technique on MATLAB. G Ajay et al [11] presented an automatic evaluation method for the determination of the quality of milled rice. Veena.H et al [12] presented an automatic evaluation method for determination of quality of milled rice. An automated system is introduced which is used for grain type identification and analysis of rice quality (i.e. Basmati, Boiled and Delhi) and grade (i.e. grade 1, grade 2, and grade3)]using Probabilistic Neural Network by Megha R. Siddagangappa et al [13].Paper proposed a new principal component analysis based approach for classification of different variety of basmati rice by Rubi Kambo et al [14]. Vidya Patil et al [15] proposed a work where image processing technique was used as an attempt to automate the process which overcomes the drawbacks of manual process. This paper provides the quality assessment of rice grains based on its size.

Ref. Index	Features Used	Tech Used	No. of Image	Success Rate
1.	Major & Minor Axis Length, Area	Not clearly Mentioned	105	93%
2.	Major & Minor Axis Length, aspect ratio, Orientation, Area, Perimeter	Measure Parameter Value	1509	Not Mentioned
3.	Perimeter, Area, Major & Minor Axis Length	Based on Feature	Not Mentioned	Not Mentioned
4.	Length, Width, Perimeter Compactness ratio	Morphological Processing	95	Not Mentioned
5.	Area, Major & Minor Axis Length	Neural Network	20	91.3%
6.	Length, Eccentricity, Major & Minor Axis Length	Based on Dimension	22	98.7%
7.	Major & Minor Axis Length, Area, Eccentricity, Perimeter	Mahalanobis Distance	40	Not Mentioned
8.	Area, Major & Minor Axis Length, Centroid	Based on Size	100	Not Mentioned
9.	Height, Width, Area, Mean, Convex Area, Angle, Perimeter, Angle_back, Eccentricity, Standard Deviation.	Image Processing	2700	Not Done yet

Table 2.1: Comparison Table

Chapter 3

Architecture of System

An architecture of proposed rice grain identification system is shown in Fig 1. An objective of the project is to design a rice grain classification and identification system using its morphological features, which classifies the species of rice grains.

ALGORITHM :

Input: Rice sample input image.
Output: Classified grain quality grade.

Fig. 3.1 : Steps of algorithm:

```
Step1: Pre-process the images of rice to remove background noise
Step2: Convert the pre processed image to binary image.
Step3: Region label the binary image. .
Step4: Segment/crop the individual grains present in the image.
Step5: Extract the geometric features major axis, minor axis and area of all the individual grains
Step6: Perform analysis on the quality using the average values of the features extracted
Step7: Classify the sample for the Type and grade based on the analysis
Stop.
```

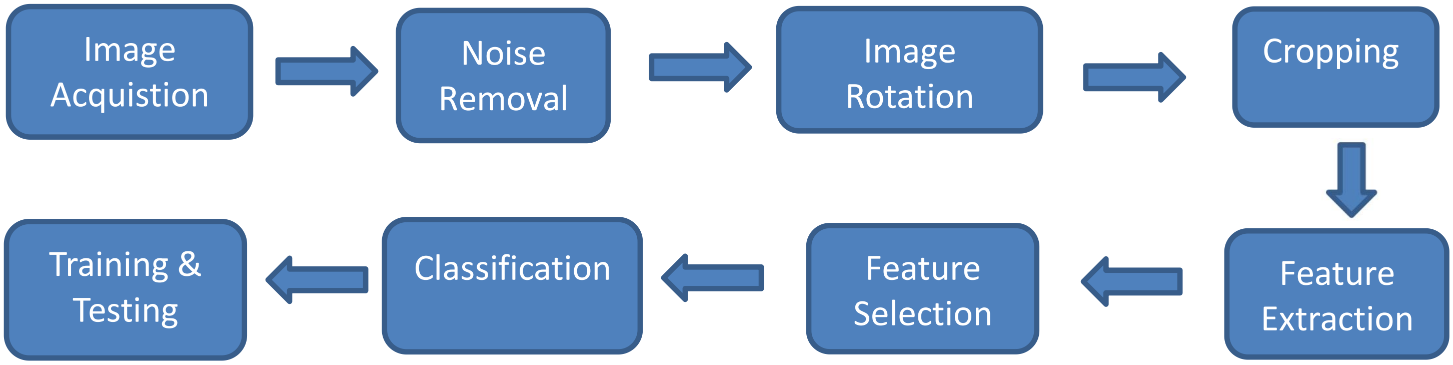


Fig. 3.1

Fig. 3.2 : Different Phases of proposed system

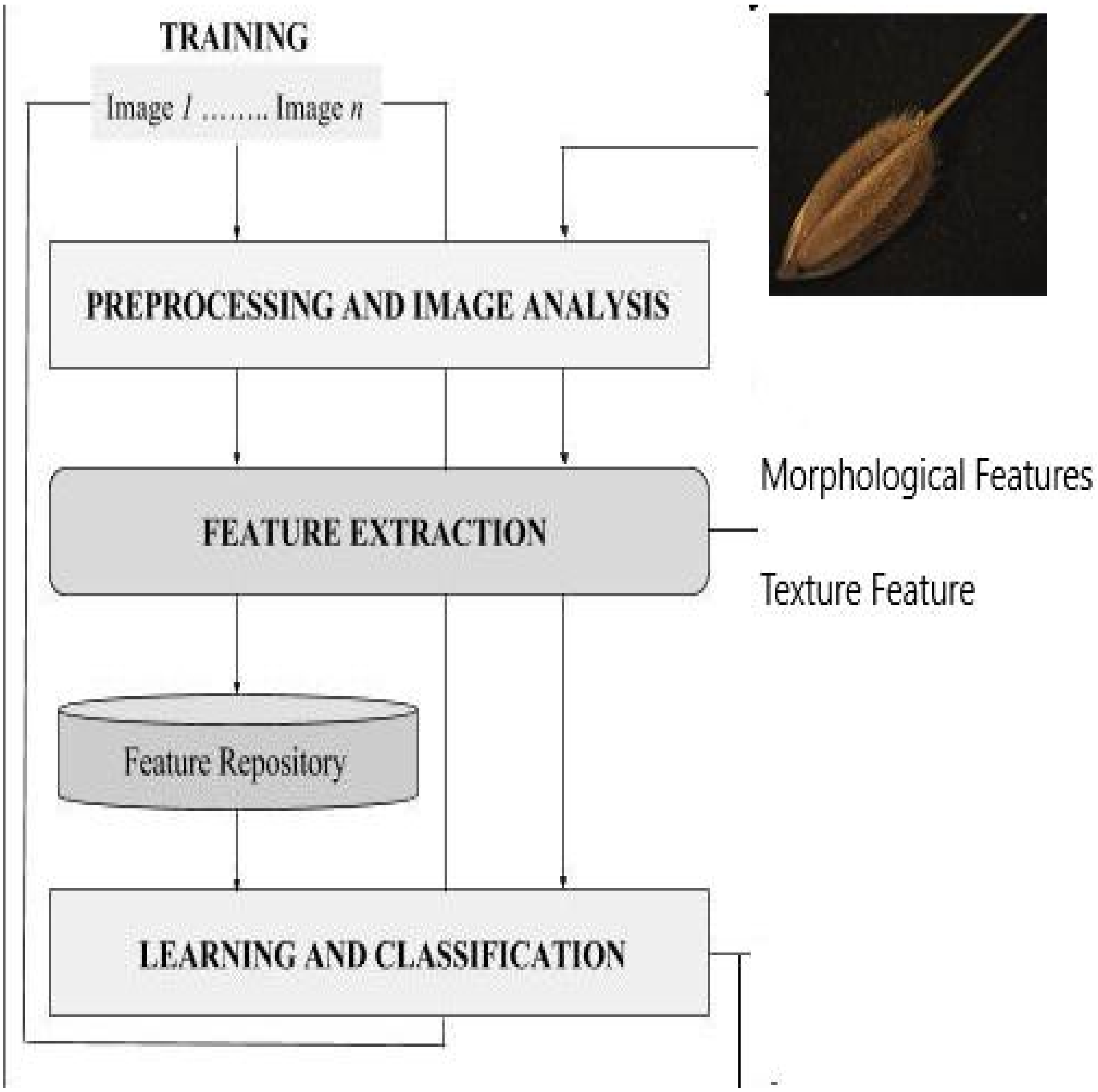


Fig. 3.2

Chapter 4

Preprocessing

4.1 Image Acquisition:

A Tucsen (USB 2.0 H Series), Model No-ISH500, S/N: KC500404063 colour camera was used to record the images of the rice of the different size. The camera was mounted on a stand which provided easy vertical movement and stable support. When the camera was fixed the distance between the lens and the sample table with uniform background, was 8.5 mm. The background was a black. The uniform intensity of lighting on the sample table was provided. The Model No of Light Source is ZEISS CL.1500 HAL. The sample seeds were collected from Bidhan Chandra Krishi Viswa Vidyalaya. Each seeds were arranged in the Horizontal orientation and position inside the field of view. The images of the 27 individual rice species of mixed varieties and 100 images of different varieties were taken for analysis. The File Format of the image in jpg, preview resolution is 1272*952 and file size is 165 KB.

4.2 Image pre-processing :

The first problem encountered during the whole process was the Image had black backgrounds, So the black background has to be removed. All of the rice grains were not in parallel with the horizontal plane. So the rice grains had to be rotated. So the aim of pre-processing is an improvement of image data that suppresses unwanted distortion or enhances some image features for further processing. For human viewing, Image Enhancement improves the classification and clarity of images. Removing noise and blur, rising contrast and enlightening details from images are examples of enhancement operation.

The aim of pre-processing is an improvement of image data that suppresses unwanted distortion or enhances some image features for further processing. For human viewing, Image Enhancement improves the quality and clarity of images. Removing noise and blur, rising contrast and enlightening details from images are example of enhancement operation.Pre-processing stepscarried out in this work are using matlab tools:

4.2.1 Gray scale conversion :

Grey scale-digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. = rgb2gray(RGB) converts the true color image RGB to the grayscale intensity image I. rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

4.2.2 Labelling of regions :

In order to perform a pattern recognition approach each isolatedregion is labelled using the command 'bwlabel'.L = bwlabel(BW, n) returns a matrix L, of the same size as BW, containing labels for the connected objects in BW. Function bwconncomp() [CC = bwconncomp(BW)] returns the connected components CC found in BW. The binary image BW can have any dimension. CC is a structure with four fields. bwconncomp uses a default connectivity of 8 for two dimensions.

4.2.3 Image segmentation :

First of all the image is converted into RGB images. Hence after converting the input image in Red, Green and Blue plane it can be seen that the red plane is clearer than the other models. So, after the image segmentation into red, blue and green planes the best one is chosen based on the clarity of the rice grain on the image. The red one is chosen because of the clarity of the rice grain in the red channel. After that the red channel image is binarized to get the rice grain as black and the background as white, after generating the black and white image a lot of noises can we observed.

The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

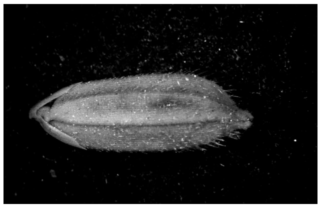


Fig. 4.1 :Red Plane

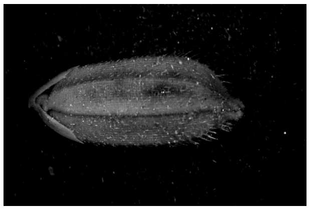


Fig 4.2: Green Plane

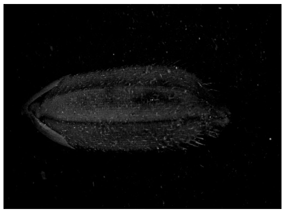


Fig 4.3: Blue Plane

Grain image is extracted from the whole image.

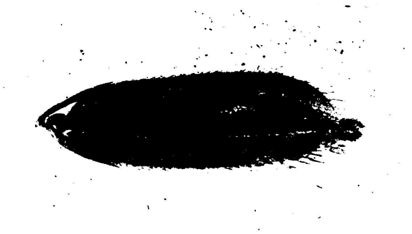


Figure 4.4: Binary Image with noise

Figure 4.5: Binary Image without noise

4.2.4 Background subtraction :

It is known as foreground detection where an image foreground(generally an image region of interest are object in its foreground) is extracted for further processing. So to do this the connected component analysis is done and the biggest area pixel is chosen among them. So after that the rice grain is extracted. White background is cropped out and based on the coordinates the main rice grain image is extracted from the whole image.

When we were trying to remove the noise some of the portion of rice grain were detected as noise and got erased that is why we stored both the image, one with background and other is without background so that we can cross check the accuracy of our results for both the images.

Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a graylevel or a color (color labeling) according to the component it was assigned to.

Extracting and labeling of various disjoint and connected components in an image is central to many automated image analysis applications.

Connected component labeling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values V. (For a binary image V={1}; however, in a graylevel image V will take on a range of values, for example: V={51, 52, 53, ..., 77, 78, 79, 80}.)

Connected component labeling works on binary or graylevel images and different measures of connectivity are possible. However, for the following we assume binary input images and 8-connectivity. The connected components labeling operator scans the image by moving along a row until it comes to a point p (where p denotes the pixel to be labeled at any stage in the scanning process) for which V={1}. When this is true, it examines the four neighbors of p which have already been encountered in the scan (i.e. the neighbors (i) to the left of p, (ii) above it, and (iii and iv) the two upper diagonal terms). Based on this information, the labeling of p occurs as follows:

- If all four neighbors are 0, assign a new label to p, else
- if only one neighbor has V={1}, assign its label to p, else
- if more than one of the neighbors have V={1}, assign one of the labels to p and make a note of the equivalences

After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels might be different graylevels or colors.

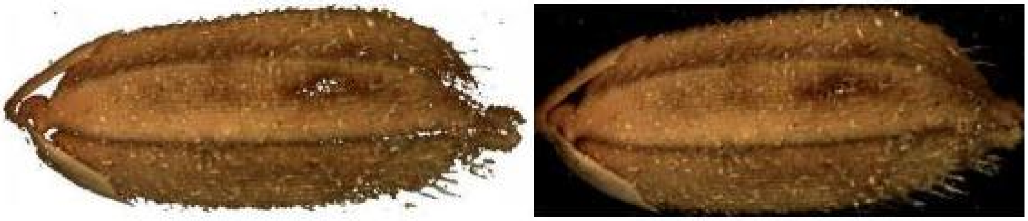


Fig 4.6

4.2.5. Image orientation :

Rotate an Image To rotate an image, use the imrotate function. When you rotate an image, you assign the image to be rotated and the rotation angle, in degrees. If you assign a positive rotation angle, imrotate rotates the image counterclockwise; if you assign a negative rotation angle, imrotate rotates the image clockwise. By default, the output image is large enough to include the entire original image. Pixels that fall outside the boundaries of the original image are set to 0 and appear as a black background in the output image. You can, however, specify that the output image be the same size as the input image, using the 'crop' argument. By default, imrotate uses nearest-neighbor interpolation to determine the value of pixels in the output image, but you can specify other interpolation methods.

This images are rotated and stored for the further processing of the rice grain. After that two coordinates of the ending point of rice grain is taken out. The angle is calculated corresponding to the horizontal plane. The angle calculated is the angle based on which the rice grain is again rotated so that it becomes parallel to the horizontal plane.

Chapter 5

Feature Extraction

5.1 System Phases

Our proposed system consists of an image analysis phase and feature extraction phase.

Image analysis involves certain processing techniques including image transformation and segmentation. It is a ‘preparatory’ phase which is vital for optimum feature extraction.

Feature extraction is necessary for forming a feature or pattern vector. This vector contains information which helps the classifier in distinguishing different types of diseased fundus images. Accurate features imply better classification.
Feature extraction is the process of extracting pieces of information or features from an image and/or object which helps us to predict whether an image is diseased or non-diseased. Feature extraction provides data to the machine learning algorithms and carried out on both the training and the testing datasets. Here the authors have considered 3 types of features namely the statistical features, texture-based features and disease specific features

5.1.1 Morphological Features :

There are two kind of feature extracted morphological and texture feature. Morphological parameters were extracted by edge detection from binary image and regionprops() function. The parameters extracted are area, height, width, eccentricity, front and back angle, perimeter, convex area and solidity and they are as follows :

- The **Height and width** of an object are diameters (lines through the center) of it. The height is the longest diameter and the width the shortest. For example, the following image is explained in terms of major and minor axis for height and width respectively.

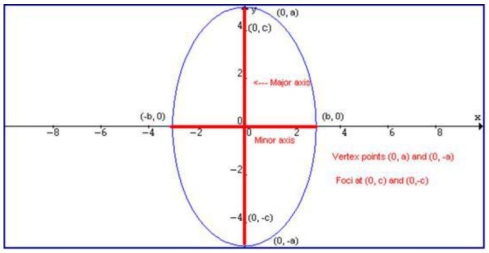


Fig. 5.1

- The **eccentricity** of an ellipse is a measure of how nearly circular the ellipse. Eccentricity is found by the following formula,

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

where a is the major axis length and b is the minor axis length of a ellipse shaped object.

- A **perimeter** is a surrounding path in a two-dimensional shape. It is the the length of the outline of a shape which is also called its circumference.

- A **convex area** is basically a region where, for every pair of points within the region and the points on the straight line joining two points also lies within the region. The boundary of such convex set is called a convex curve and the area bounded is called convex area.

- The **angle front** is the angle made by the tangents at the starting point of the object. It is mainly an acute angle with the less than 90 degrees.

- The **angle back** is the angle made by the tangents at the end point of the object. It is mainly an acute angle with the less than 90 degrees.

- **Solidity** is the ratio of area of an image to its convex area.

- **Area** is the total number of pixel contained in an image.

- **Centroid** is the center of mass of an image, where horizontal coordinate is known as x-centroid and vertical coordinate is known as y-centroid.

- Another very common shape factor is the **circularity** (or isoperimetric quotient), a function of the perimeter *P* and the area *A:*

$$f_{\mathrm{circ}} = \frac{4\pi A}{P^2}$$

The circularity of a circle is 1, and much less than one for a starfish footprint. The reciprocal of the circularity equation is also used, such that f_{circ} varies from one for a circle to infinity.

- Another very common shape factor is the **rectangularity**, a function of the perimeter *P* and the area *A:*

Rectangularity =Area/Perimeter

- **Extent** is the proportion of the pixels in the bounding box that are also in the region. Computed as the Area divided by the area of the bounding box.

- **The length of minor axis** is given by the formula :

$$Minor\ axis = \sqrt{[(a+b)^2 - f^2]}$$

Where *f* is the distance between foci and *a,b* are the distances from each focus to any point on the ellipse.

- **The length of major axis is** given by the formula :

$$Major\ axis = a+b$$

Where *a,b* are the distances from each focus to any point on the ellipse.

- **Awn** is a hairy, or bristle-like, appendage growing from the ear or flower or barley, rye and many types of widely growing grasses.

$$Awn = width - major\ axis$$

5.1.2 Texture Feature

Texture is one of the most important defining characteristic of image. It is characterized by the special distribution of Gray levels in a neighbourhood.

- **Energy** is a minimization or maximization problem, it depends on the work we need. It is also used in object detection or segmentation tasks.

$$\sum_{i=1}^{k}\sum_{j=1}^{k} p_{ij}{}^{2}$$

- **Correlation** is a measure of how correlate a pixel is to its neighbour over the whole image.

$$\sum_{i,j}\frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$$

- **Contrast** is a measure of the intensity between a pixel and its neighbour over the whole image.

$$(x-t)_{+} = \left\{\begin{matrix} x-t \\ 0, \end{matrix}\right. \text{ if } x > t, \text{otherwise}$$

and

$$(t-x)_{+} = \left\{\begin{matrix} t-x \\ 0, \end{matrix}\right. \text{ if } x < t, \text{otherwise}$$

- **Homogeneity** measures the closeness of the distribution of pixels in a Gray scale image.

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\frac{p_{ij}}{1+|i-j|}$$

- The **standard deviation** is a measure that is used to calculate the amount of variation or dispersion in the set of data values. It is given by the formula :

$$\mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

- The **mean** is basically the average of the total numbers. To calculate mean, just sum up all the numbers and then divide it by the number of total numbers added.

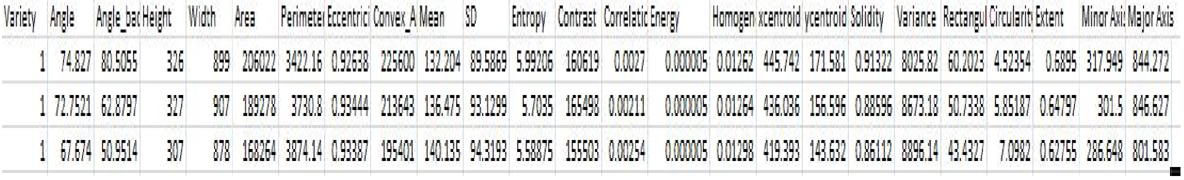
$$\bar{x} = \frac{\sum x}{N}$$

$$\sum x = \text{ the sum of } x$$

$$N = \text{ number of data}$$

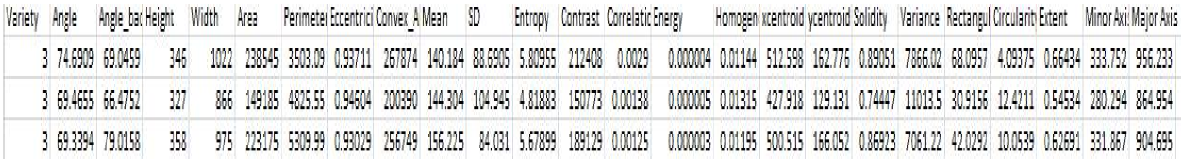
5.2 Dataset Creation

The following Dataset is obtained from MATLAB. We have taken 100 images of each rice variety.



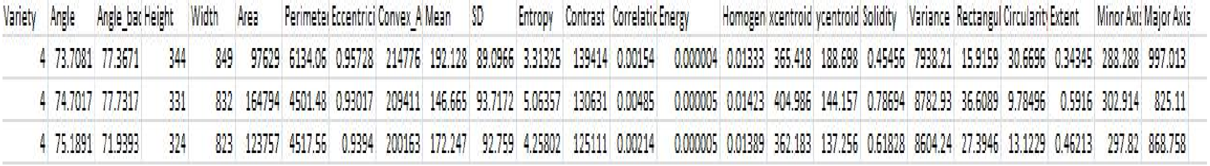
Variety	Angle	Angle_bac	Height	Width	Area	Perimeter	Eccentrici	Convex_A	Mean	SD	Entropy	Contrast	Correlatic	Energy	Homogen	xcentroid	ycentroid	Solidity	Variance	Rectangul	Circularit	Extent	Minor Axi	Major Axis
1	74.827	80.5055	326	899	206022	3422.16	0.92638	225600	132.204	89.5869	5.99206	160619	0.0027	0.000005	0.01262	445.742	171.581	0.91322	8025.82	60.2023	4.52354	0.6895	317.949	844.272
1	72.7521	62.8797	327	907	189278	3730.8	0.93444	213643	136.475	93.1299	5.7035	165498	0.00211	0.000005	0.01264	436.036	156.596	0.88596	8673.18	50.7338	5.85187	0.64797	301.5	846.627
1	67.674	50.9514	307	878	168264	3874.14	0.93387	195401	140.135	94.3193	5.58875	155503	0.00254	0.000005	0.01298	419.393	143.632	0.86112	8896.14	43.4327	7.0982	0.62755	286.648	801.583

Fig. 5.2 : Dataset of Danti



Variety	Angle	Angle_bac	Height	Width	Area	Perimeter	Eccentrici	Convex_A	Mean	SD	Entropy	Contrast	Correlatic	Energy	Homogen	xcentroid	ycentroid	Solidity	Variance	Rectangul	Circularit	Extent	Minor Axi	Major Axis
3	74.6909	69.0459	346	1022	238545	3503.09	0.93711	267874	140.184	88.6905	5.80955	212408	0.0029	0.000004	0.01144	512.598	162.776	0.89051	7866.02	68.0957	4.09375	0.66434	333.752	956.233
3	69.4655	66.4752	327	866	149185	4825.55	0.94604	200390	144.304	104.945	4.81883	150773	0.00138	0.000005	0.01315	427.918	129.131	0.74447	11013.5	30.9156	12.4211	0.54534	280.294	864.954
3	69.3394	79.0158	358	975	223175	5309.99	0.93029	256749	156.225	84.031	5.67899	189129	0.00125	0.000003	0.01195	500.515	166.052	0.86923	7061.22	42.0292	10.0539	0.62691	331.867	904.695

Fig. 5.3 : Dataset of Harnana



Variety	Angle	Angle_bac	Height	Width	Area	Perimeter	Eccentrici	Convex_A	Mean	SD	Entropy	Contrast	Correlatic	Energy	Homogen	xcentroid	ycentroid	Solidity	Variance	Rectangul	Circularit	Extent	Minor Axi	Major Axis
4	73.7081	77.3671	344	849	97629	6134.06	0.95728	214776	192.128	89.0966	3.31325	139414	0.00154	0.000004	0.01333	365.418	188.698	0.45456	7938.21	15.9159	30.6696	0.34345	288.288	997.013
4	74.7017	77.7317	331	832	164794	4501.48	0.93017	209411	146.665	93.7172	5.06357	130631	0.00485	0.000005	0.01423	404.986	144.157	0.78694	8782.93	36.6089	9.78496	0.5916	302.914	825.11
4	75.1891	71.9393	324	823	123757	4517.56	0.9394	200163	172.247	92.759	4.25802	125111	0.00214	0.000005	0.01389	362.183	137.256	0.61828	8604.24	27.3946	13.1229	0.46213	297.82	868.758

Fig. 5.4 : Dataset of Sabraj

Chapter 6

Feature Selection

6.1 Spearman Algorithm

The Spearman's rank-order correlation is the nonparametric version of the Pearson product-moment correlation. Spearman's correlation coefficient, (ρ, also signified by r_s) measures the strength and direction of association between two ranked variables.

You need two variables that are either ordinal, interval or ratio (see our Types of Variable guide if you need clarification). Although you would normally hope to use a Pearson product-moment correlation on interval or ratio data, the Spearman correlation can be used when the assumptions of the Pearson correlation are markedly violated. However, Spearman's correlation determines the strength and direction of the **monotonic relationship** between your two variables rather than the strength and direction of the linear relationship between your two variables, which is what Pearson's correlation determines . There are two methods to calculate Spearman's correlation depending on whether: (1) your data does not have tied ranks or (2) your data has tied ranks. The formula for when there are no tied ranks is:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where d_i = difference in paired ranks and n = number of cases. The formula to use when there are tied ranks is:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

where i = paired score

6.2 Feature Selected

- Angle from front
- Angle from Back
- Area
- Eccentricity
- Mean
- Entropy
- Contrast
- Homogeneity
- Y Centroid
- Solidity
- Rectangularity
- Circularity
- Minor Axis
- Awn
- Extent

Chapter 7

Classification

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. In order to carry out classification, we require a training set of "features" which have been extracted from the image or provided directly along with each image. The training set is also labeled with the known value of the grade. Once training is complete, a portion of the dataset (which was not used for training) is tested with the learned model and the output is compared with the ground truth label in order to determine the accuracy of the model. The process, as a whole, consists of three phases: feature extraction, training, and testing.

Once we have obtained our dataset consisting of the feature vectors of our images, we move on to the classification phase. Here, the dataset is divided into two subsets: a training set and a testing set. The training set, as its name implies, is used to 'train' or 'teach' the classifier. It includes both the feature vector of each image and its corresponding class. The testing set is used to measure the performance of the trained classifier. This set consists of only the feature vectors. The result of classification on testing set is measured against the actual class values and the overall classification error is computed. This gives us an estimated error, or performance, of our system.

For preliminary analysis, we have considered two different classifiers each having its own advantages and disadvantages. They are briefly described in the subsections that follow

7.1 The Random Forests Classifier

Random Forest Classifier is an ensemble classifier or regressor which is a group of multitude of decision trees to be used in the training data set. The result of the training set is the mode of the class or classes predicted by the constituent decision trees.

An ensemble method is the one which works by divide and conquer policy. By an ensemble classifier, a group of weak learners can become a strong learner for better performance. Each weak learner is capable of learning from the datasets and predicting some target value for the testing set.

The performance of classifiers largely depends upon the quality of training dataset used during the learning purpose. However, one common misconception worldwide is the larger the training set, the better the performance. This is not true. The performance of the classifier is dependent on the quality of dataset and not upon its size. The quality of the dataset includes proper demarcations between feature values of different classes and little or no ambiguity. But the identification of good dataset values from the training set is not feasible. So, Random forest relies upon the fact that if different subsets of the training data be fed into different classifiers, then the cumulative performance of the classifiers can be improved to a good extent. Often, the ensemble method uses a weighted mean of the predictions of individual classifier to specify a target value. The figure below would help in greater understanding of the ensemble classifier.

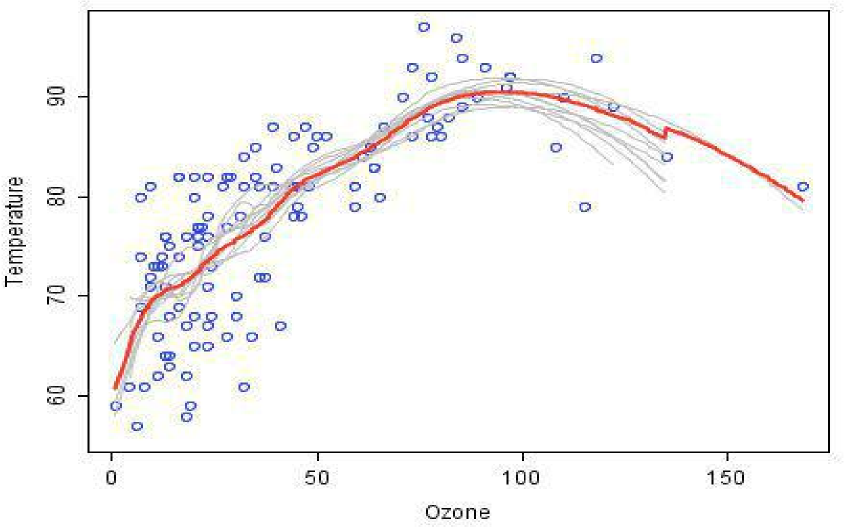


Fig. 7.1. A plot of classifiers under random forest.

In Fig. 7.1, the gray lines show the weak learners , that is the individual trees and the red line shows the strong learner which gives a much more accurate result.

Decision Trees

Decision trees have a predictive model that uses observations about an item to predict its target values. The input enters the tree through the root and traverses down the tree based on certain conditions or criteria. The leaves in such a tree, represent the classes or labels for prediction.

The number of conditions is equal to the level of the tree and the number of leaf nodes represents the number of classes.

For example, if there is to be a decision to be made about playing a game, depending on several variables like, outlook, humidity, etc, then following diagram helps in making the right decision depending upon the preferences provided by the user.

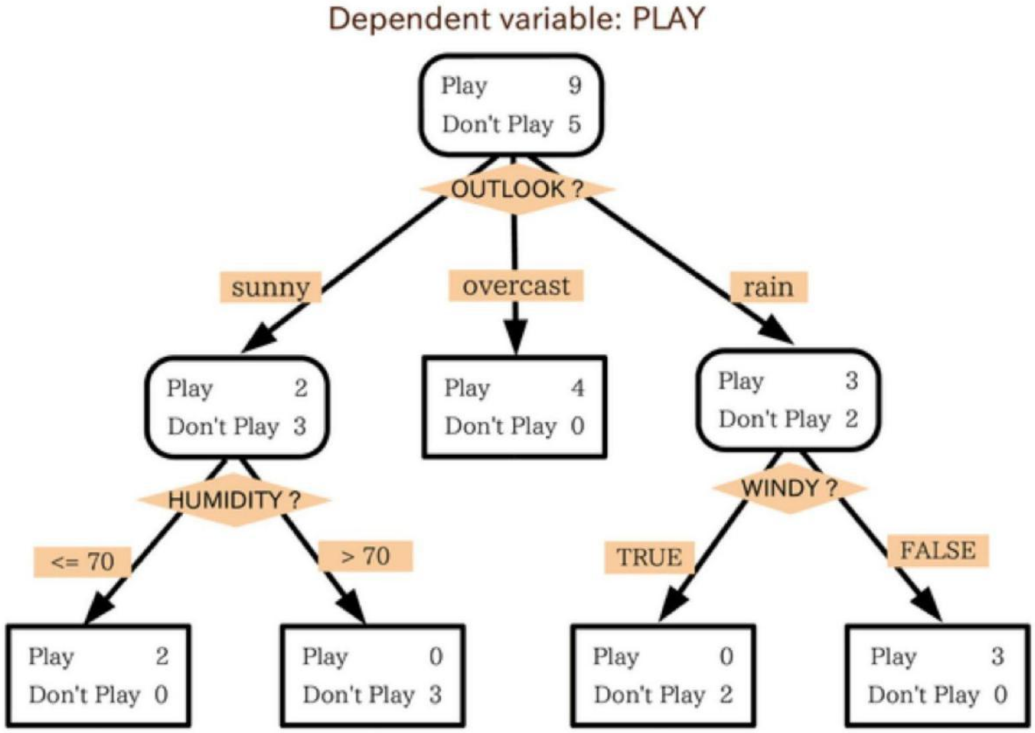


Fig. 7.2 A decision tree.

The random forest is a combination of all such decision trees.

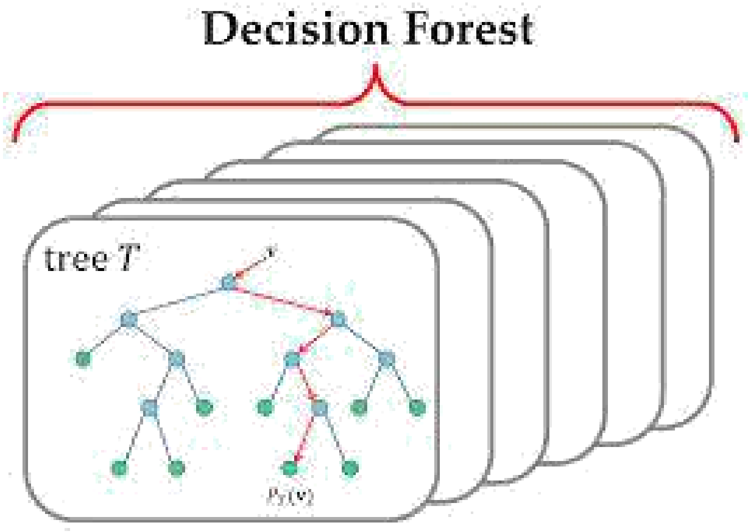


Fig. 7.3. Random forest as an ensemble.

Here is how such a system is trained; for some number of trees T:

1. N cases are sampled at random with replacement to create a subset of the data . The subset should be about 66% of the total set.
2. At each node:
   a) For some number m, m predictor variables are selected at random from all the predictor variables.
   b) The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.
   c) At the next node, choose another m variables at random from all predictor variables and do the same.

   Depending upon the value of m, there are three slightly different systems:

1. Random splitter selection: m =1
2. Breiman's bagger: m = total number of predictor variables
3. Random forest: m << number of predictor variables. Brieman suggests three possible values for m: ½√m, √m, and 2√m

When a new input is entered into the system, it is run down all of the trees. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in the case of categorical variables, a voting majority

It is to be noted that:

1. With a large number of predictors, the eligible predictor set will be quite different from node to node.
2. The greater the inter-tree correlation, the greater the random forest error rate, so one pressure on the model is to have the trees as uncorrelated as possible.
3. As m goes down, both inter-tree correlation and the strength of individual trees go down. So some optimal value of m must be discovered.

The figures below helps in understanding the above-mentioned fundamentals of random forest classifier.

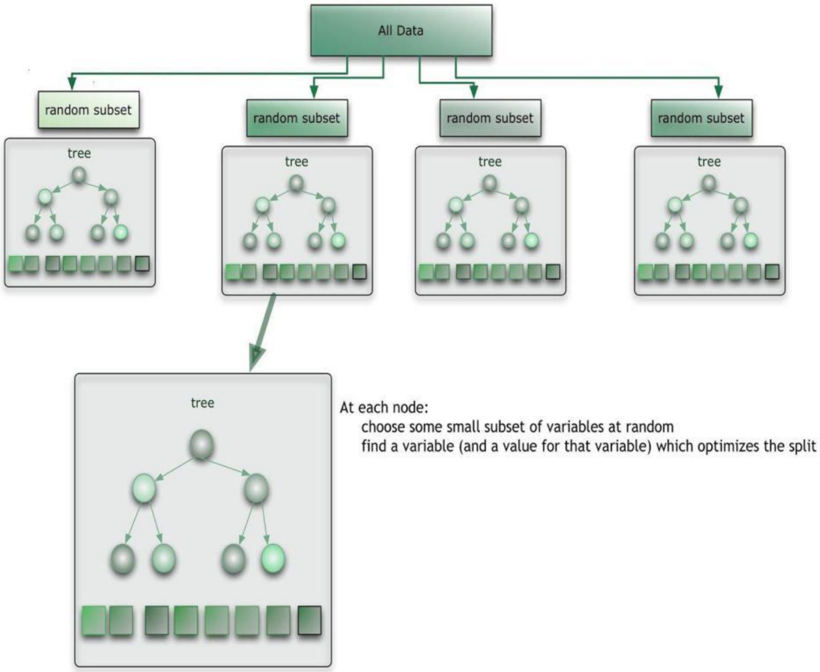

All Data
random subset
random subset
random subset
random subset
tree
tree
tree
tree
tree
At each node:
choose some small subset of variables at random
find a variable (and a value for that variable) which optimizes the split

Fig. 7.4 Working of a random forest classifier.

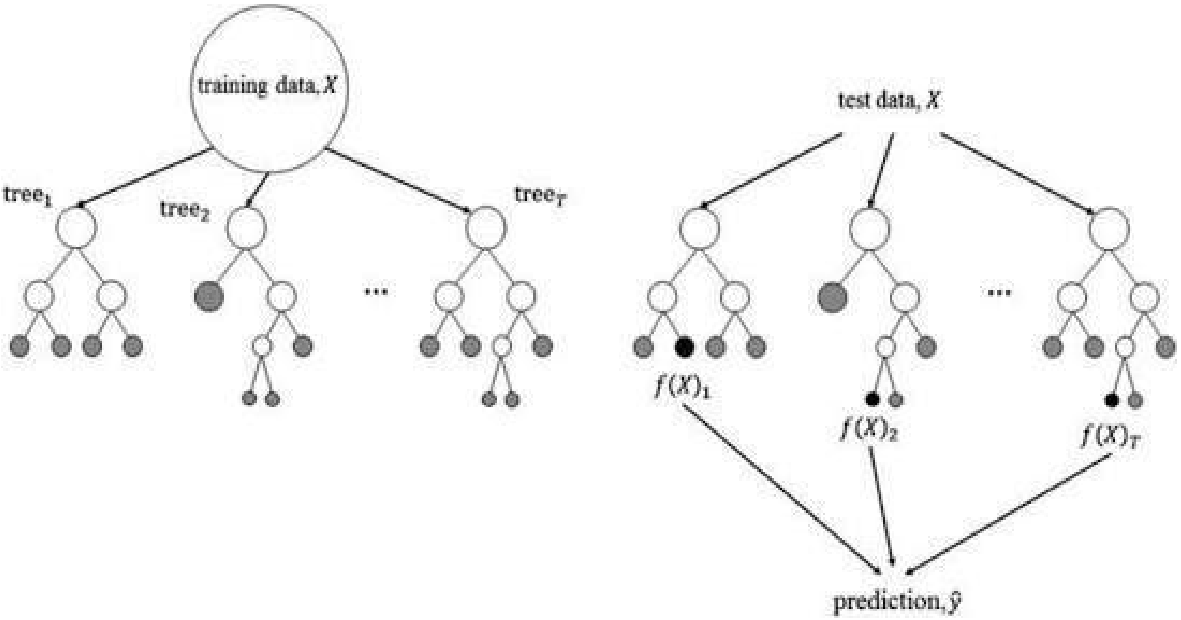


Fig. 7.5 Training and testing phase of Random forest classifier.

Strengths of Random Forest Classifier :

1. Random Forests work very fast as compared to most other classifiers.
2. They can work upon unbalanced data and the probability of error is reduced.
3. The RF does not require all feature vector values to be specified. It can work on missing data as well.

Drawbacks of Random Forest Classifier :

1. When used as a regressor, Random forest cannot predict a value beyond the range available to it in the training data.
2. Over-fitting is sometimes a problem in these classifiers, particularly when the dataset is noisy.

7.2 The Feed-forward Artificial Neural Network (ANN) with Backpropagation

Artificial Neural Network

Artificial neural networks are a machine learning technique modeled on biological neural networks in the human brain. As in the brain, an ANN consists of layers of interconnected 'neurons' that accept a set of features as input and output either a class or a continuous variable depending on whether the network is used for classification or regression. A typical network consists of a single input layer which accepts the input features for the training or testing example (as the case may be), one or more 'hidden' layers, and a single output layer that produces the final result.

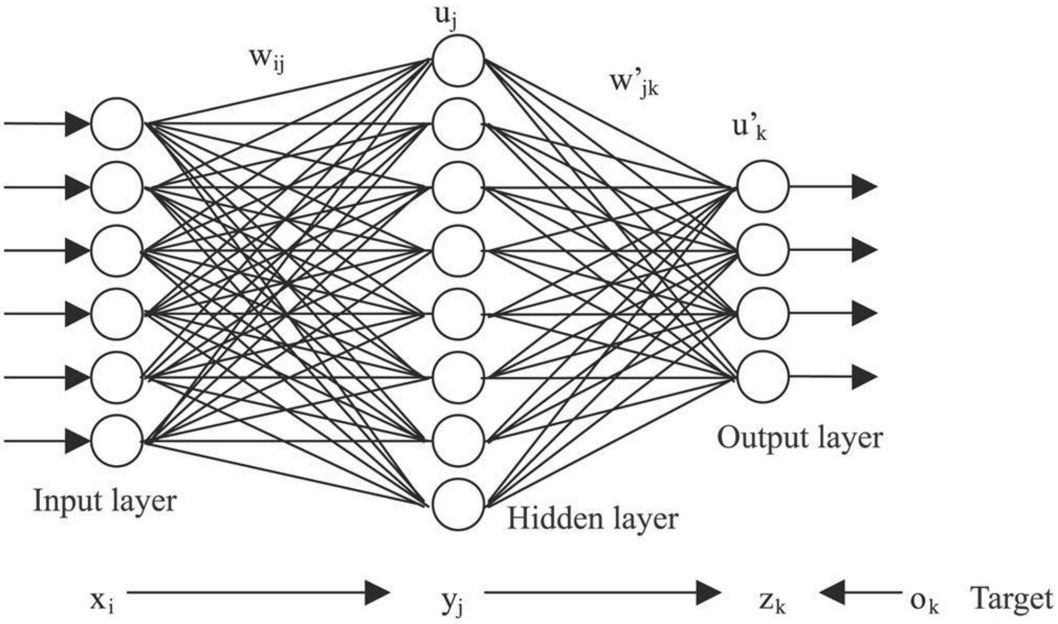


Fig. 7.6 An Artificial neural network with three layers.

An ANN is typically defined by three parameters:

1. The interconnection pattern between the different layers of neurons: this determines the type of network. A simple feed forward network involves forward interconnections between two consecutive layers in which the weighted inputs propagate through the network in the forward direction until they reach the output layer where the final result is obtained. In a recursive neural network, however, feedback loops exist that connect the output layer with previous input or hidden layers.

2. The weights of the interconnections or links which are updated during the learning process. Different learning algorithms such as gradient descent, conjugate gradient, and Newton's method are used to adjust and learned the appropriate weights that will result in a generalized function that will appropriately approximate the training set.

3. The activation function that converts a neuron's weighted input into its output activation. Activation functions will be explained in more detail shortly.

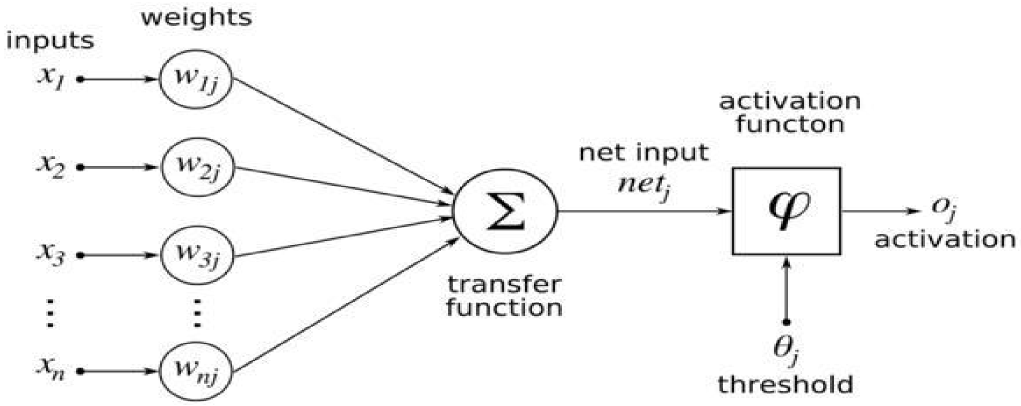


Fig. 7.7 A single neuron with its inputs and activation function.

The figure above depicts the j th neuron in the first layer of the network. Here the input features consist of the terms x1 , x2 till xn . Each xi term is multiplied by its corresponding weight for the j th neuron, i.e., wi j . The weighted features are then summed up to produce the net input term net j . This term is then used as input for the activation function φ of the neuron. For all layers excluding the output layer, the output oj of the activation function serves as the j th input feature for the next layer.
In case of ANNs used for regression, the output layer consists of a single neuron whose continuous output is the predicted value for the set of input features. For ANNs used for classification, the number of neurons in the output layer is the same as the number of classes with each neuron producing a continuous, real value as before. However, the difference in this case lies in the fact that we now choose the class whose corresponding output neuron produces the largest value - generally the softmax function is used for this purpose.

Activation functions

The activation function of a neuron, as stated before, determines the real-valued output for that neuron based on the weighted inputs supplied to it. There exists several desirable properties for activation functions:

1. ***Nonlinear*** in order to universally generalize functions.
2. ***Continuously differentiable*** so that gradient optimization methods such as gradient descent may be used during the learning process.
3. ***Range*** - a finite range will increase stability of gradient based optimization while an infinite range will increase efficiency and will require a lower learning rate.
4. ***Monotonic*** so that the error surface is guaranteed to be convex.
5. ***Smooth*** in order to better generalize functions as per experimental observations.

Approximates identity near the origin so that the network weights can be initialized with small random values. If the activation function does not have this property, then special care must be taken when initializing the weights.

Functions commonly used as activation functions include:

1. Unit step (Heaviside)

$$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$$

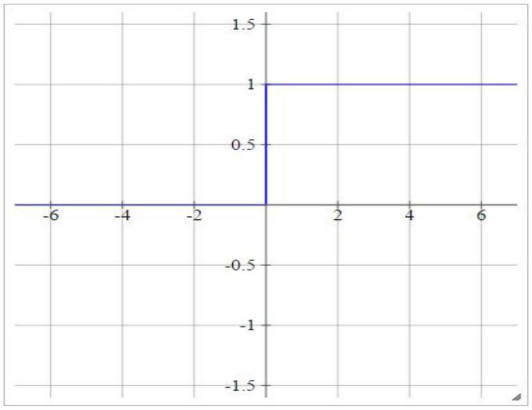


Fig. 7.8 Plot of the Heaviside function

2. Sign (Signum)

$$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$$

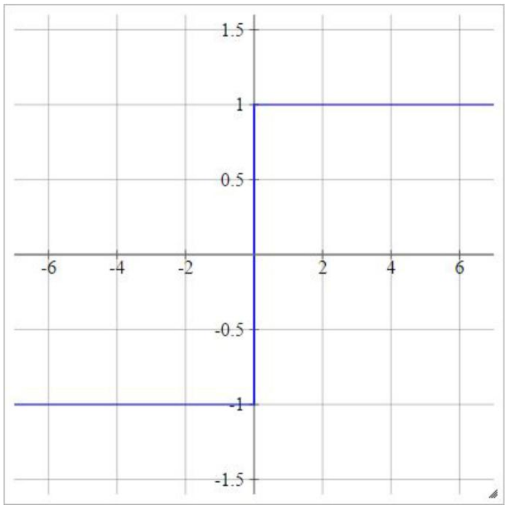


Fig. 7.9 Plot of the sign function

3. Linear

$$\phi(z) = z$$

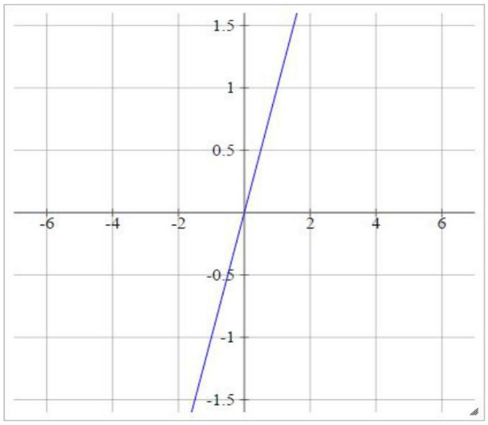


Fig. 7.10 Plot of the linear function.

4. Logistic (sigmoid)

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

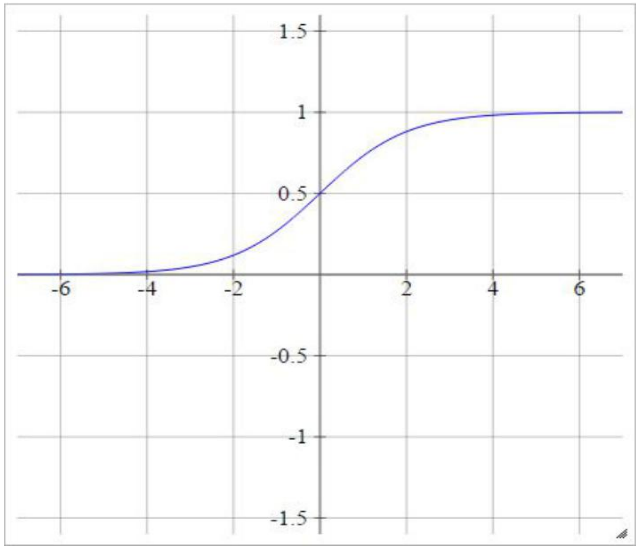


Fig. 7.11 : Plot of the sigmoid function.

5. Hyperbolic tangent

$$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

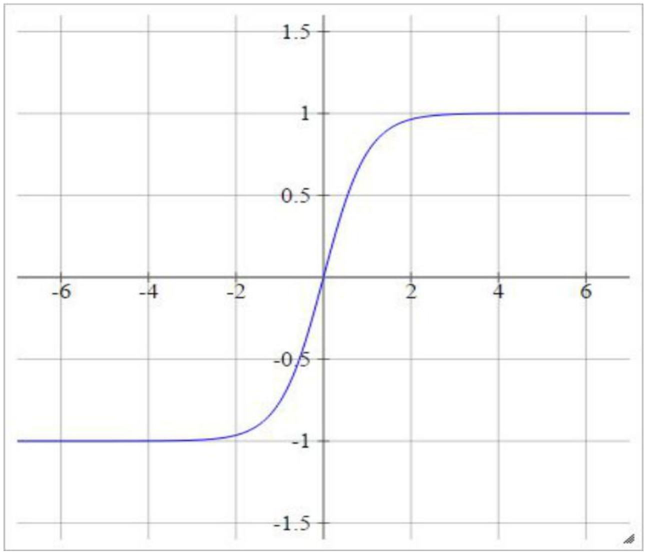


Fig. 7.12 : Plot of the hyperbolic tangent function.

Backpropagation

The backpropagation algorithm is a very popular method which is used in classification using neural networks. It is based on the concept of "propagating" the error obtained using a set of weights "backwards" through the neural network. The network then tries to adjust, and hence "learn", the correct weights by minimising the error obtained.

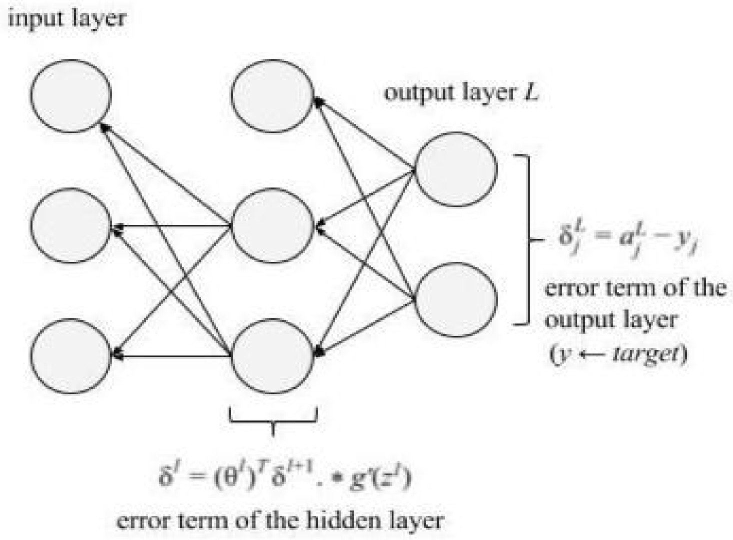

input layer
output layer L
$\delta_j^L = a_j^L - y_j$
error term of the output layer
($y \leftarrow target$)
$\delta^l = (\theta^l)^T \delta^{l+1} . * g'(z^l)$
error term of the hidden layer

Fig. 7.13 : The backpropagation mechanism.

The error in the output layer is obtained easily by simply finding the difference between the output layer values and the actual training set target values. Thereafter, for each layer ***i,*** we compute its error in terms of the error obtained in layer ***i+1*** . The algorithm makes use of the following equations representing the error terms:

(i) $\delta_j^L = a_j^L - y_j$

(ii) $\delta^l = (\theta^l)^T \delta^{l+1} . * g'(z^l)$ where $g'(z^l) = a^l . * (1 - a^l)$

Here, L denotes the output layer, l denotes a hidden layer, θ^l is a vector containing the weights for layer l , a^l represents the activation values of neurons in layer l and δ^l is the error vector for layer l .In brief, the algorithm involves four main steps:

Algorithm

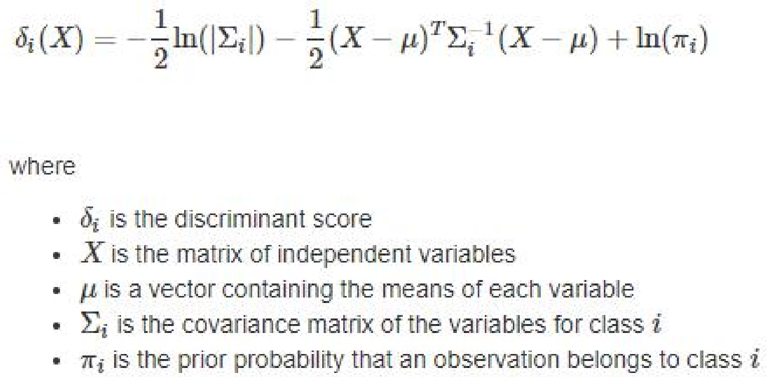
```
1.  For  each feature vector:
2.      Use the weights to carry out a simple feed forward propagation
3.      Compute the error terms using equations (i) and (ii)

4.      Adjust the weights in order to minimise the error terms as far as possible, or until
        the error satisfies a specified accuracy
5.  End
```

Each iteration of the above steps resulting in a weight adjustment is known as a learning epoch. For the purpose of obtaining even better results with backpropagation, a technique known as Bayesian regularization has been used for the learning phase. Regularization may be described as the process of introducing additional information to prevent overfitting and increase generalization of the network. Bayesian regularization, in particular, minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. This allows us to use a higher number of hidden neurons to obtain even better accuracy

7.3 Quadratic Discriminant Analysis

A **quadratic classifier** is used in machine learning and statistical classification to separate measurements of two or more classes of objects or events by a quadric surface. It is a more general version of the linear classifier/

The QDA classifier assumes that the observations from each class of *Y* are drawn from a Gaussian distribution. QDA assumes that each class has its own covariance matrix. It finds a quadratic function of the independent variables. Mathematically, it assumes that an observation from the *k*th class is of the form X~ N(μk,Σk), where Σk is a covariance matrix for the *k*th class. Under this assumption, the classifier assigns an observation to the class for which discriminant screre is largest . Below is the formula to calculate discriminant scores.



$$\delta_i(X) = -\frac{1}{2}\ln(|\Sigma_i|) - \frac{1}{2}(X-\mu)^T\Sigma_i^{-1}(X-\mu) + \ln(\pi_i)$$

where

- δ_i is the discriminant score
- X is the matrix of independent variables
- μ is a vector containing the means of each variable
- Σ_i is the covariance matrix of the variables for class i
- π_i is the prior probability that an observation belongs to class i

Looking at the data it is clear that the variability of the observations within each class differ. QDA is able to capture the differing covariances and provide more accurate non-linear classification decision boundaries.

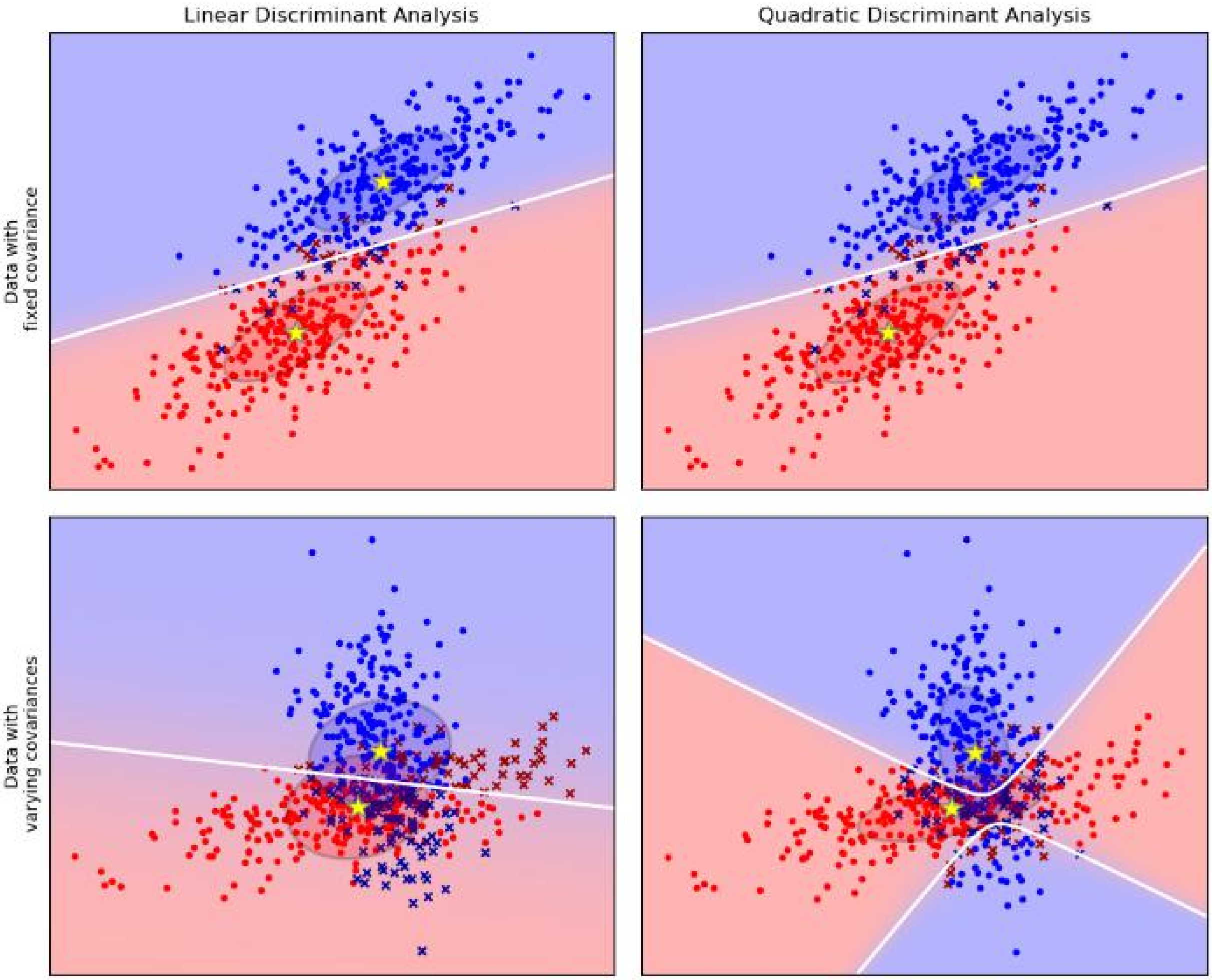


Fig. 7.14 : Comparison between LDA and QDA

Discriminant analysis is used to determine which variables discriminate between two or more naturally occurring groups. For example, an educational researcher may want to investigate which variables discriminate between high school graduates who decide (1) to go to college, (2) NOT to go to college. For that purpose the researcher could collect data on numerous variables prior to students' graduation. After graduation, most students will naturally fall into one of the two categories. Discriminant Analysis could then be used to determine which variable(s) are the best predictors of students' subsequent educational choice. Computationally, discriminant function analysis is very similar to analysis of variance (ANOVA). For example, suppose the same student graduation scenario. We could have measured students' stated intention to continue on to college one year prior to graduation. If the means for the two groups (those who actually went to college and those who did not) are different, then we can say that the intention to attend college as stated one year prior to graduation allows us to discriminate between those who are and are not college bound (and this information may be used by career counselors to provide the appropriate guidance to the respective students). The basic idea underlying discriminant analysis is to determine whether groups differ with regard to the mean of a variable, and then to use that variable to predict group membership (e.g. of new cases).

Discriminant Analysis may be used for two objectives: either we want to assess the adequacy of classification, given the group memberships of the objects under study; or we wish to assign objects to one of a number of (known) groups of objects. Discriminant Analysis may thus have a descriptive or a predictive objective. In both cases, some group assignments must be known before carrying out the Discriminant Analysis. Such group assignments, or labeling, may be arrived at in any way. Hence Discriminant Analysis can be employed as a useful complement to Cluster Analysis (in order to judge the results of the latter) or Principal Components Analysis.

7.4 Multivariate Adaptive Regression Spline

Multivariate Adaptive Regression Spline is a tool used for statistical analysis and in data mining.It is a non-parametric regression technique and can be seen as an extension of linear models that automatically models non-linearities and interaction between models. Data is partitioned 60% as training samples and 40% as testing samples. The multi-label data contains attributes with multiple classes which are converted into single label classes. Each class of the target variable and its attributes are given as input to MARS then it creates the model and graphically displays collision of each predictive aspect on the outcome. MARS builds the model in the form:

$$\hat{f}(x)=\sum_{m=1}^{k} c_m B_m(x)$$

where, k is a number of variables,**Bm(x)** is a basic function and **Cm** is a constant coefficient. Basic function has any one of the three forms: a) a constant 1 or b) hinge function or c) product of more than two hinge function. Hinge function may be **max(0,x-c)** or **max(0,c-x)** . The forward stepwise process is stopped when some maximum model size is reached. The backward pruning procedure is applied to the model removing the least important term one at a time the best fitting model in a stepwise sequence is chosen with the fit measured by generalized cross validation(GCV) criterion.

GCV helps to compare the performance of the model , creates subsets and chooses the best subset features.After selecting the features combine all the feature of each class sort the feature according to the weight of the variable.

The effectiveness of MARS model is from the

The piecewise linear basis function which permit fast updating of least squares fit as the knot position is changed.

The hinge function automatically partition the input data so the effect of outliers is contained.

It tend to have good bias variance trade off.

Earth models are similar to but not identical to models built by other MARS implementations. The differences stem from the forward pass where small implementation differences (or perturbations of the input data) can cause somewhat different selection of terms and knots (although similar GRSq's).
The result of the forward pass is the MARS basis matrix bx and the set of terms defined by dirs and cuts (these are all fields in earth's return value, but the bx returned by the forward pass includes all terms before trimming back to selected.terms).

The forward pass adds terms in pairs until the first of the following conditions is met: (i) Reached the maximum number of terms nk (ii) Adding a term changes RSq by less than 0.001 (iii) Reached a RSq of 0.999 or more (iv) GRSq is less than -10 (a pathologically bad GRSq, FAQs 12.12 and 12.13) (v) Reached numerical accuracy limits (no new term increases RSq).

The backward passes give identical or near identical results, given the same forward pass results.

The pruning pass (also called the backward pass) is handed the set of terms bx created by the forward pass. Its job is to find the subset of those terms that gives the lowest GCV. The following description of the pruning pass explains how various fields in earth's returned value are generated. The pruning pass works like this: it determines the subset of terms in bx (using pmethod) with the lowest RSS (residual sum-of-squares) for each model size in 1:nprune. It saves the RSS and term numbers for each such subset in rss.per.subset and prune.terms. It then calculates the GCV with penalty for each entry of rss.per.subset to yield gcv.per.subset. Finally it chooses the model with the lowest value in gcv.per.subset, puts its term numbers into selected.terms, and updates bx by keeping only the selected.terms.

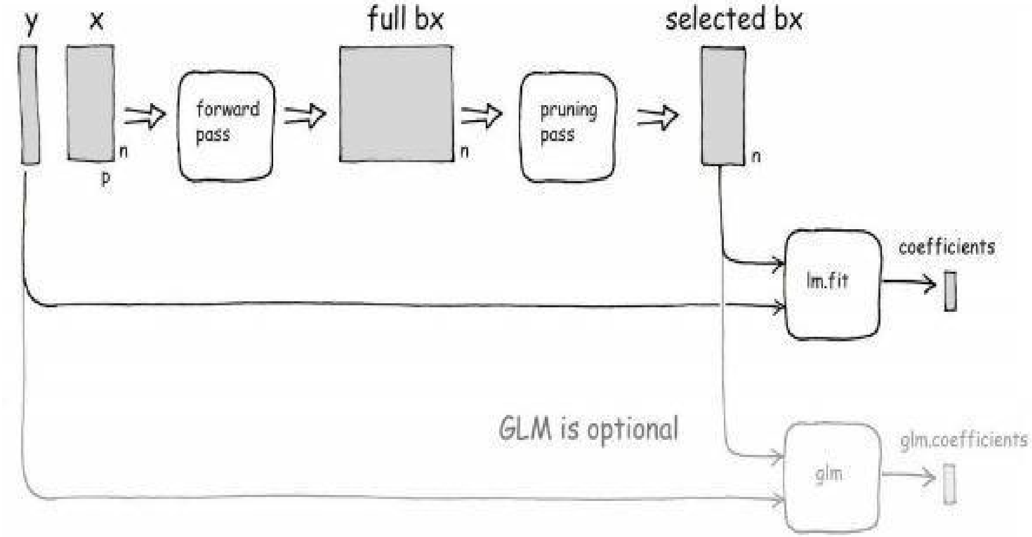


Fig. 7.15 : MARS Model

After the pruning pass, earth runs lm.fit to determine the fitted.values, residuals, and coefficients, by regressing the response y on bx. This is an ordinary least-squares regression of the response y on the basis matrix bx (see Figure 1 and example (model.matrix.earth) for an example). If y has multiple columns then lm.fit is called for each column. If a glm argument is passed to earth, earth runs glm on (each column of) y in addition to the above call to lm.fit. Set trace >= 3 to trace the pruning pass.

Generalized cross validation (GCV)

GCV = RSS / (N * (1 – EffectiveNumberOfParameters / N)^2)

– where RSS is the residual sum-of-squares measured on the training data and N is the number of observations (the number of rows in the x matrix).

– The EffectiveNumberOfParameters is defined in the MARS context as

– EffectiveNumberOfParameters = NumberOfMarsTerms + Penalty * (NumberOfMarsTerms

– 1) / 2

– where Penalty is about -1 (the EARTH software allows the user to preset Penalty).

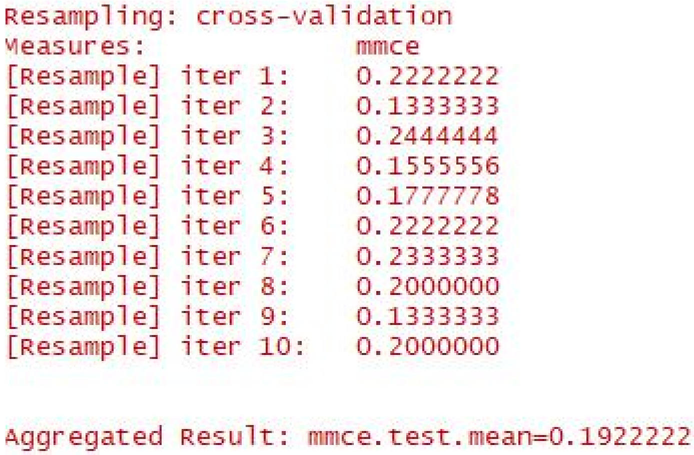
Chapter 8

Results & Discussions

This section illustrate the accuracy and evolution matrix for the selected features of multileveled dataset. The rice variety dataset contains 18 features and 900 objects of 9 varieties.

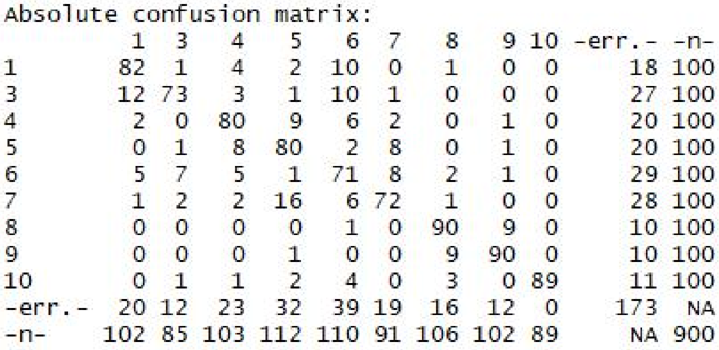
8.1 Random Forest

The results obtained from Random Forest are as follows:



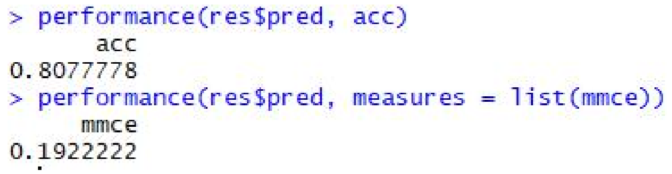
```
Resampling: cross-validation
Measures:             mmce
[Resample] iter 1:    0.2222222
[Resample] iter 2:    0.1333333
[Resample] iter 3:    0.2444444
[Resample] iter 4:    0.1555556
[Resample] iter 5:    0.1777778
[Resample] iter 6:    0.2222222
[Resample] iter 7:    0.2333333
[Resample] iter 8:    0.2000000
[Resample] iter 9:    0.1333333
[Resample] iter 10:   0.2000000


Aggregated Result: mmce.test.mean=0.1922222
```

Fig. 8.1 : Random Forest - 10 Cross Validation



```
Absolute confusion matrix:
           1  3   4   5   6  7   8   9 10 -err.- -n-
1         82  1   4   2  10  0   1   0  0     18 100
3         12 73   3   1  10  1   0   0  0     27 100
4          2  0  80   9   6  2   0   1  0     20 100
5          0  1   8  80   2  8   0   1  0     20 100
6          5  7   5   1  71  8   2   1  0     29 100
7          1  2   2  16   6 72   1   0  0     28 100
8          0  0   0   0   1  0  90   9  0     10 100
9          0  0   0   1   0  0   9  90  0     10 100
10         0  1   1   2   4  0   3   0 89     11 100
-err.-    20 12  23  32  39 19  16  12  0    173  NA
-n-      102 85 103 112 110 91 106 102 89     NA 900
```

Fig. 8.2 : Random Forest - Confusion Matrix

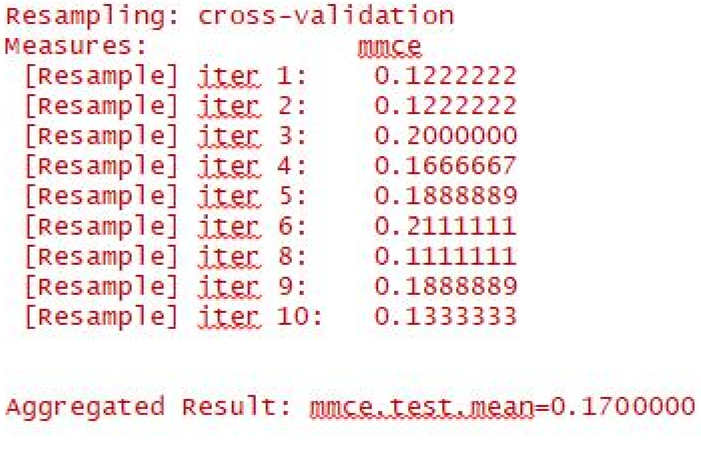


```
> performance(res$pred, acc)
      acc
0.8077778
> performance(res$pred, measures = list(mmce))
     mmce
0.1922222
```

Fig. 8.3 : Random Forest - Accuracy

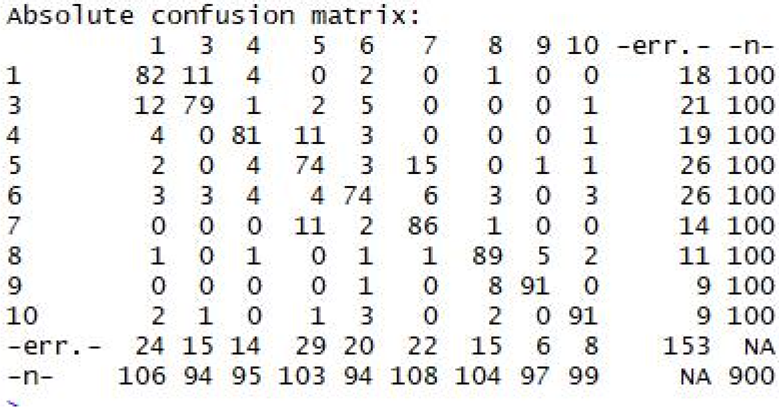
8.2 Neural Network

The results obtained from Neural Network are as follows:



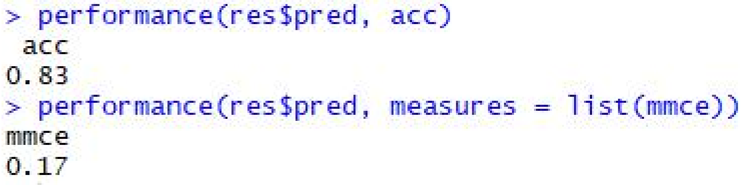
```
Resampling: cross-validation
Measures:                mmce
 [Resample] iter 1:     0.1222222
 [Resample] iter 2:     0.1222222
 [Resample] iter 3:     0.2000000
 [Resample] iter 4:     0.1666667
 [Resample] iter 5:     0.1888889
 [Resample] iter 6:     0.2111111
 [Resample] iter 8:     0.1111111
 [Resample] iter 9:     0.1888889
 [Resample] iter 10:    0.1333333


Aggregated Result: mmce.test.mean=0.1700000
```

Fig. 8.4 : Neural Network - 10 Cross Validation



```
Absolute confusion matrix:
          1  3  4   5  6   7   8  9 10 -err.- -n-
1        82 11  4   0  2   0   1  0  0     18 100
3        12 79  1   2  5   0   0  0  1     21 100
4         4  0 81  11  3   0   0  0  1     19 100
5         2  0  4  74  3  15   0  1  1     26 100
6         3  3  4   4 74   6   3  0  3     26 100
7         0  0  0  11  2  86   1  0  0     14 100
8         1  0  1   0  1   1  89  5  2     11 100
9         0  0  0   0  1   0   8 91  0      9 100
10        2  1  0   1  3   0   2  0 91      9 100
-err.-   24 15 14  29 20  22  15  6  8    153  NA
-n-     106 94 95 103 94 108 104 97 99     NA 900
```

Fig. 8.5 : Neural Network - Confusion Matrix

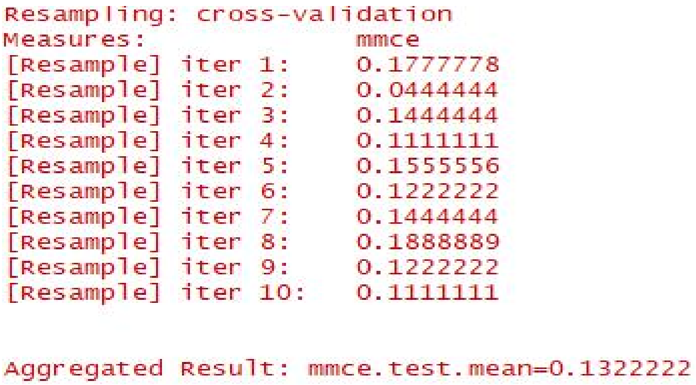


```
> performance(res$pred, acc)
 acc
0.83
> performance(res$pred, measures = list(mmce))
mmce
0.17
```

Fig. 8.6 : Neural Network - Accuracy

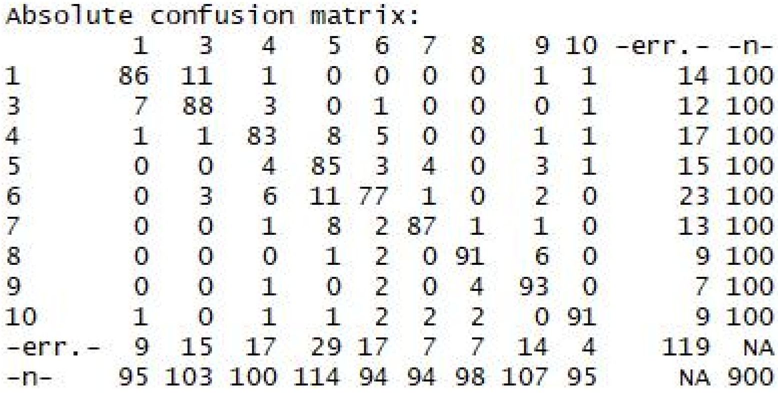
8.3 Quadratic Discrimination Analysis

The results obtained from QDA are as follows:



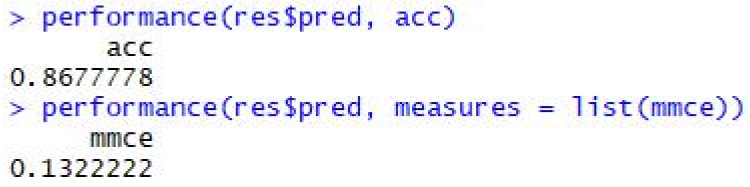
```
Resampling: cross-validation
Measures:             mmce
[Resample] iter 1:    0.1777778
[Resample] iter 2:    0.0444444
[Resample] iter 3:    0.1444444
[Resample] iter 4:    0.1111111
[Resample] iter 5:    0.1555556
[Resample] iter 6:    0.1222222
[Resample] iter 7:    0.1444444
[Resample] iter 8:    0.1888889
[Resample] iter 9:    0.1222222
[Resample] iter 10:   0.1111111


Aggregated Result: mmce.test.mean=0.1322222
```

Fig. 8.7 : QDA - 10 Cross Validation



```
Absolute confusion matrix:
         1   3   4   5  6  7  8   9 10 -err.- -n-
1       86  11   1   0  0  0  0   1  1     14 100
3        7  88   3   0  1  0  0   0  1     12 100
4        1   1  83   8  5  0  0   1  1     17 100
5        0   0   4  85  3  4  0   3  1     15 100
6        0   3   6  11 77  1  0   2  0     23 100
7        0   0   1   8  2 87  1   1  0     13 100
8        0   0   0   1  2  0 91   6  0      9 100
9        0   0   1   0  2  0  4  93  0      7 100
10       1   0   1   1  2  2  2   0 91      9 100
-err.-   9  15  17  29 17  7  7  14  4    119  NA
-n-     95 103 100 114 94 94 98 107 95     NA 900
```

Fig. 8.8 : QDA - Confusion Matrix

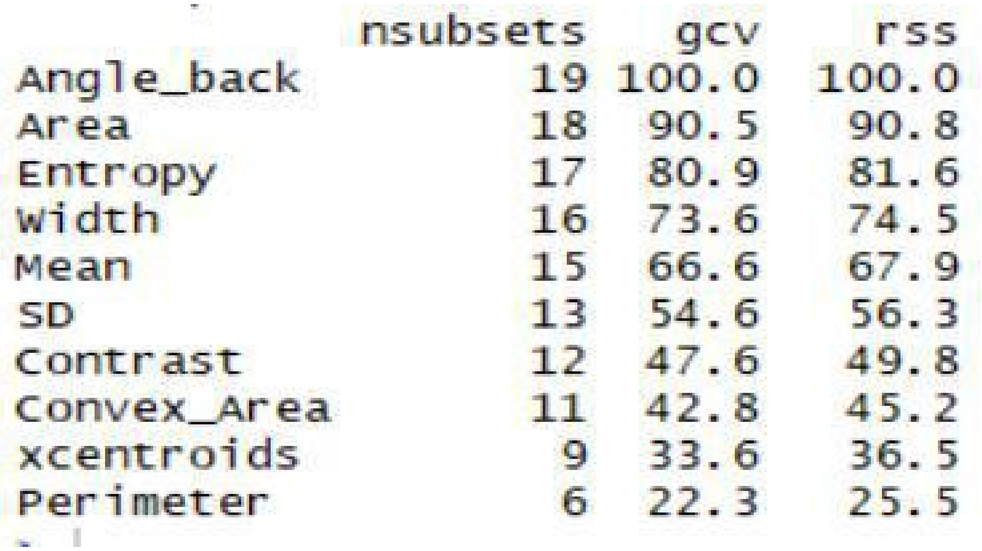


```
> performance(res$pred, acc)
      acc
0.8677778
> performance(res$pred, measures = list(mmce))
     mmce
0.1322222
```

Fig. 8.9 : QDA - Accuracy

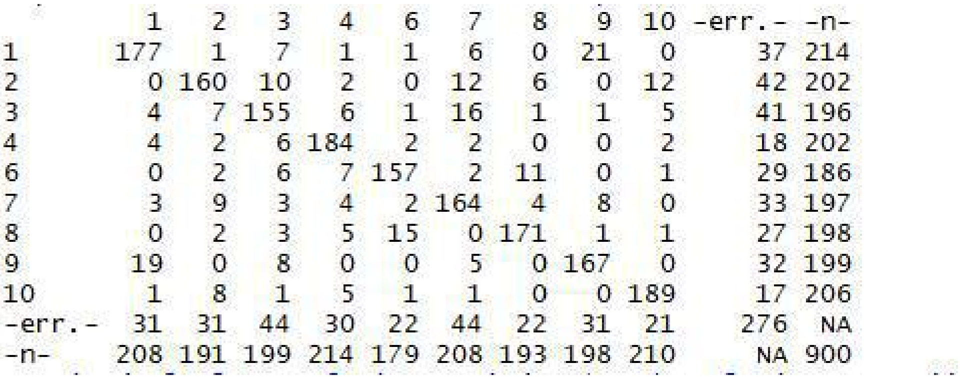
8.4 Multivariete Adoptive Regression Spline

The results obtained from MARS are as follows:



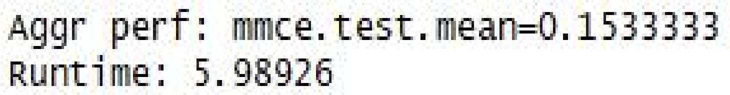
```
              nsubsets   gcv   rss
Angle_back          19 100.0 100.0
Area                18  90.5  90.8
Entropy             17  80.9  81.6
Width               16  73.6  74.5
Mean                15  66.6  67.9
SD                  13  54.6  56.3
Contrast            12  47.6  49.8
Convex_Area         11  42.8  45.2
xcentroids           9  33.6  36.5
Perimeter            6  22.3  25.5
```

Fig. 8.10 : - 10 Best Features Selected using MARS



```
          1   2   3   4   6   7   8   9  10 -err.- -n-
1       177   1   7   1   1   6   0  21   0     37 214
2         0 160  10   2   0  12   6   0  12     42 202
3         4   7 155   6   1  16   1   1   5     41 196
4         4   2   6 184   2   2   0   0   2     18 202
6         0   2   6   7 157   2  11   0   1     29 186
7         3   9   3   4   2 164   4   8   0     33 197
8         0   2   3   5  15   0 171   1   1     27 198
9        19   0   8   0   0   5   0 167   0     32 199
10        1   8   1   5   1   1   0   0 189     17 206
-err.-   31  31  44  30  22  44  22  31  21    276  NA
-n-     208 191 199 214 179 208 193 198 210     NA 900
```

Fig. 8.11 : MARS - Confusion Matrix



```
Aggr perf: mmce.test.mean=0.1533333
Runtime: 5.98926
```

Fig. 8.12 : MARS - Accuracy

In each of the classification methods considered below, we used a set of 900 images of rice grain. Of these 900 images, a fraction were used for training the classifier and the remaining images were used for testing. The overall accuracies obtained for each classifier are shown in Table 1 below.

Classifier	Max. Accuracy
Neural Net	83 %
Quadratic Discrimintant Analysis	86.78%
Multivariate Regression Splines	84.7 %
Random Forest	80.77%

Table 8.13 : Performance data for the neural network and random forest classifiers

Ref. Index	Features Used	Tech Used	No. of Image	Success Rate
1.	Major & Minor Axis Length, Area	Not clearly Mentioned	105	93%
2.	Area, Major & Minor Axis Length	Neural Network	20	91.3%
3.	Length, Eccentricity, Major & Minor Axis Length	Based on Dimension	22	98.7%
4.	Morphological & Texture Feature (Proposed Method)	Image Processing & Soft Computing	900 (9 variety)	86.78%

Table 8.14 : Comparative study of proposed method with prevailing method

Chapter 9

Conclusion

The processing of imagery and the vigilant assortment of the variety measured in this effort for extracting features from rice granules significantly abridged the intricacy of the grading problem

In this project report, we have worked with various morphological processes on rice species and extracting features using image processing. This method is computationally efficient with more accurate result than the other existing methods. Using Matlab we efficiently worked with various inbuilt functions which help to get expected outcomes. All this lead to better classify and identify rice grains

Chapter 10

Reference

[1] Brosnan T and Sun D W (2002) Inspection and grading of agricultural and food products by computer
[2]
vision systems – a review. Computers and Electronics in Agriculture 36:193-213.
[2] Camelo GA (2012) et al Digital image analysis of diverse Mexican rice cultivars .Journal of the Science of Food and Agriculture 92:2709-2714.

[3] Gujjar H S and Siddappa M (2013) A Method for identification of basmati rice grain of India and its
quality using pattern classification. International Journal of Engineering Research and Applications 3:268-273.

[4] Guzman J D and Peralta E K (2008) Classification of Philippine rice grains using machine vision and
artificial neural networks. World Conference on Agricultural Information and IT 19:41-48.

[5] Kaur G, Din S, Brar A S and Singh D (2014) Scanner image analysis to estimate leaf area. International Journal of Computer Applications 107:5-10.

[6] Kaur H, Singh B (2013) Classification and grading rice using multi-class svm. International Journal
of Scientific and Research Publications 3:1-5.

[7] Liu Z Y, Cheng F J, Ying Y B and Rao X Q (2005) Identification of rice seed varieties using neural
network. Journal of Zhejiang University Science 11: 1095-1100.

[8] Maheshwari C V, Jain K R and Modi C K (2012) Nondestructive quality analysis of Indian Gujrat-17
oryza sativa ssp indica (rice) using image processing. International Journal of Computer Engineering Science 2:48-54.

[9] Patil N K, Malemath V S and Yadahalli R M (2011) Color and texture based identification and classification of food grains using different color models and haralick features. International Journal on
Computer Science and Engineering 3:3669-3680.

[10] Shantaiya S and Ansari U (2010) Identification of food grains and its quality using pattern classification. International Journal of Computer &amp; Communication Technology 2:70-74.

[11] Singh T, Kumar CM, Singh P and Kumar P (2013) Advances in computer vision technology for foods of animal and aquatic origin- a review. Journal of Meat Science and Technology 1:40-49.

[12]. Neelamegam, P., Abirami, S.Vishnu Priya, K.Rubalya Valantina, S." Analysis of Rice Granules using
Image Processing and Neural Network Pattern Recognition Tool" Information &amp; Communication
Technologies (ICT), IEEE Conference ,879 - 884,11-12 April 2013.

[13] Sukhvir Kaur, Derminder Singh ." Geometric Feature Extraction of Selected Rice Grains using Image
Processing Techniques" International Journal of Computer Applications (0975 – 8887)Volume 124 – No.8, August 2015. \

[14] Abirami. S, Neelamegam. P, Kala. H "Analysis of Rice Granules using Image Processing and Neural
Network Pattern Recognition Tool" International Journal of Computer Applications (0975 – 8887) Volume 96– No.7, June 2014.

[15] Priyankaran Tanck, Bipan Kaushal ,"A New Technique of Quality Analysis for Rice Grading for Agmark Standards" International Journal of Innovative Technology and Exploring Engineering (IJITEE)
2278-3075, Volume-3, Issue-12, May 2014.

[16] G.Ajay, M.Suneel, K.Kiran Kumar, P.Siva Prasad, "Quality Evaluation of Rice Grains Using Morphological Methods" International Journal of Soft Computing and Engineering (IJSCE): 2231-2307,
Volume-2, Issue-6, January 2013.

[17] Veena.H Latharani T R, "An efficient method for classification of rice grains using Morphological
process" International Journal of Innovative Research in Advanced Engineering (IJIRAE) 2278-2311, Volume 1,Issue 1,April 2014.

[18]. Megha R. Siddagangappa, "Classification and Quality Analysis of Food Grains" IOSR Journal of
Computer Engineering (IOSR-JCE)- ISSN: 2278-0661,p-ISSN: 2278-8727, PP 01-10, Volume 16, Issue
4, Ver. III, Jul – Aug. 2014.

[19] Rubi Kambo, Amit Yerpude "Classification of Basmati Rice Grain Variety using Image Processing and
Principal Component Analysis " International Journal of Computer Trends and Technology (IJCTT) ISSN: 2231-5381, Volume 11, number 2, May 2014.

[20] Vidya Patil, V. S. Malemath," Quality Analysis and Grading of Rice Grain Images" International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), Volume 3, Issue 6, June 2015.

[21]Xue-Bin LI, Xiao-Ling YU,"Influence of Sample Size on Prediction of Animal Phenotype Value Using Back-Propagation Artificial Neural Network with Variable Hidden Neurons" IEEE conference, 2009.

[22] http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm

[23] B. S. Anami, V. Burkpalli, S. A. Angadi, and N. M. Patil, "Neural network approach for grain classification and gradation," Proceedings of the second national conference on document analysis and
recognition, pp. 394-408, July 2003.

[24] N. S. Visen, J. Paliwal, D. S. Jayas, and N. D. G. White, "Image
analysis of bulk grain samples using neural networks," Canadian Biosystems Engineering, vol. 46, pp. 7.11-7.18, 2004.

[25] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," IEEE Trans. on Syst.,Man, and cybern, vol 6, pp. 610-621, 1973.