VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018, KARNATAKA.

A PROJECT REPORT

On

"Sugarcane Crop Support System"

Submitted in Partial Fulfillment for the Award of the Degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

1SG16CS016	ASHWIN RAMESH P
1SG16CS020	BINDHU M V
1SG16CS029	DIVIJ N
1SG16CS030	DIVYA K

Under the Guidance of
PROF. ANURADHA BADAGE
Assistant professor, Dept. of CSE

Department of Computer Science and Engineering
SAPTHAGIRI COLLEGE OF ENGINEERING
14/5, Chikkasandra, Hesarghatta Main Road
Bengaluru – 560057.

2019-2020

SAPTHAGIRI COLLEGE OF ENGINEERING

14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057.

Department of Computer Science and Engineering

Certificate

Certified that the Project Work entitled **"Sugarcane Crop Support System"** carried out by **ASHWIN RAMESH P (1SG16CS016), BINDHU M V (1SG16CS020), DIVIJ N (1SG16CS029), DIVYA K (1SG16CS030),** bonafide students of **Sapthagiri College of Engineering**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The project report has been approved as it satisfies the academic requirements in respect of **Project Work Phase II (15CSP85)** prescribed for the said degree.

Signature of the Guide
Prof. Anuradha Badage
Assistant professor

Signature of the HOD
Dr. Kamalakshi Naganna
Professor & Head

Signature of the Principal
Dr. H Ramkrishna
Principal

EXTERNAL EXAMINATION:

Name of the Examiners

Signature with Date

1._____

_____

2._____

_____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

I consider it is a privilege to express my gratitude and respect to all those who guided me in completion of technical seminar.

I am, grateful to thank our Principal **Dr. H Ramkrishna**, Sapthagiri College of Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

It's a great privilege to place on record my deep sense of gratitude to our beloved HOD **Dr. Kamalakshi Naganna** of Computer Science & Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

I am also grateful to thank my Technical seminar Guide **Prof. Anuradha Badage** of CSE department for her invaluable support and guidance.

I would also like to thank the teaching and non-teaching staff members who have helped me directly or indirectly during the technical seminar.

Lastly but most importantly I would like thank my family and friends for their co-operation and motivation to complete this seminar successfully.

ASHWIN RAMESH P	**1SG16CS016**
BINDHU M V	**1SG16CS020**
DIVIJ N	**1SG16CS029**
DIVYA K	**1SG16CS030**

ABSTRACT

The main wealth of the India is farming. But the damaging rate of that agricultural product is mainly through the natural disasters like floods and storm or secondary factors like the virus or bacteria that infect a plant. Matching with the long diversity of conditions under which sugarcane is grown around the world, there is wide spectrum of pests and diseases which have come to acquire a place of priority for control on regional or inter-regional bases due to agro-climatic management conditions associated with the area. In addition the susceptibility of the variety to the diseases and pests aggravates the situation and creates additive problems. Traditionally, diseases in plants were identified by experienced farmers or agricultural scientists in laboratories. The latter may be an expensive process of knowing the disease infected on the plant while, the former may be inaccurate and prone to human errors. To overcome the problem, sugarcane disease detection and prevention is required. A system is proposed wherein the farmer, just by clicking a picture of plant can detect the disease and receive related solutions for the same.

SAPTHAGIRI COLLEGE OF ENGINEERING
14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru-560057.

Department of Computer Science and Engineering

DECLARATION

We **ASHWIN RAMESH P(1SG16CS016), BINDHU M V(1SG16CS020), DIVIJ N(1SG16CS029)** and **DIVYA K(1SG16CS030),** bonafide students of **Sapthagiri College of Engineering,** hereby declare that the project entitled "**Sugarcane Crop Support System**" submitted in partial fulfilment for the award of Bachelor of Engineering in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2019-2020 is our original work and the project has not formed the basis for the award of any other degree, fellowship or any other similar titles.

Name & Signature of the Student with date

1)

2)

3)

4)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Fig 1.1: Sugarcane Agriculture in India

Agriculture forms the backbone of the Indian "Economy". Being the largest sector in India, agriculture is the source of livelihood for over 60 per cent population in the country. A farmer produces many products in his farm. He also uses many inputs to produce the products. Some of the products like cotton, jute and sugarcane are used in industries as raw materials. Sugarcane is an important crop used in sugar industries as raw material. The sugar industries process it into value added products like sugar, sucrose, glucose and even ethanol. Ethanol is considered to be an essential item to the masses. So, the Government takes various measures to control sugar prices within certain limits. In consideration of the economic and social importance of sugar industries, the researcher undertakes this study.

1.2 IMPORTANCE OF SUGARCANE

Sugarcane is an important cash crop grown in India. Sugarcane cultivation and development of sugar industry runs parallel to the growth of human civilization and is as old as

agriculture. The importance and use of sugarcane and sugar in the country's socio-economic milieu is deep rooted and immense. The present day rural economy set up sugarcane cultivation and sugar industry has been focal point for socio-economic development in 2 rural areas by mobilizing rural resources, generating employment and higher income, developing transport and communication facilities. About 7 million sugarcane farmers and large number of agricultural labourers are involved in sugar cane cultivation and ancillary activities. Apart from this, the sugar industry provides employment to 5 lakh skilled and semi-skilled workers in rural areas. Sugarcane is a perennial grass of the genus Saccharum. It is cultivated for its juice, from which sugar is processed. Most present-day commercial canes are the off springs or hybrids of the species of Saccharum Officinarum, which was developed from a wild cane species. Sugarcane cultivation requires a tropical or subtropical climate, with a minimum of 24 inches of annual rainfall. Sugarcane cultivation is propagated from cuttings have become common method of reproduction. Each cutting must contain at least one bud, and the cutting is usually planted by hand. Once planted, a strand of cane can be harvested several times. After each harvest, the cane sends up new stalks, called ratoon. Usually, each successive harvest gives a smaller yield and eventually the declining yields justify planting. Depending on agricultural practice, two to ten harvests may be possible between plantings. In India sugarcane is sold as jaggery (gur) and is also refined into sugar primarily for consumption in tea, coffee, and sweets and for the production of alcoholic beverages. Uses 3 of sugarcane include the production of sugar, molasses, rum, soda, and ethanol. The bagasse that remains after sugarcane crushing may be burned to provide both heat-used in the mill and electricity. Electricity produced typically sold to the electricity grid. It may also, because of its high cellulose content, be used as raw material for the production of paper, cardboard, etc. India is one of the largest sugarcane producers in the world, producing around 300 million tons of cane per annum in the year 2013-14. Production of the sugar is the second largest agro processing industry in the country after cotton and textiles. India is the only country that produces plantation sugar unlike other countries that produce raw or refined sugar or both. In India, sugarcane the key raw material for production of sugar and gur, planted once a year during January to March. It is the major cost driver for the production of sugar. It being an agricultural crop is subject to the unpredictable vagaries of nature, yielding either a bumper crop or a massive shortfall in its cultivation from year to year. The sugarcane growing may be broadly classified into two agro climatic regions – subtropical and tropical. The subtropical zone includes four States: Uttar Pradesh, Bihar, Punjab and Haryana. The tropical zones include five 4 States. These are: Maharashtra, Andhra Pradesh, Tamil Nadu, Gujarat and Karnataka. Mainly there are two ways of

sugarcane marketing. In the first way of marketing sugarcane is directly sold to sugar factories or mills that produce sugar. Sugar has historically been classified as an essential commodity and has been regulated across the value chain. The heavy regulations in the sector artificially impact the demand and supply forces resulting in market imbalance. Sensing this problem, since 1993 the regulations have been progressively eased. The key regulatory milestones include de-licensing of the industry in 1998 and removal of control on storage and distribution in 2002. Each sugar mill has a command area which binds cane farmers and sugar mills to sell and buy from each other. Sugar mills have to purchase all the cane sold to them, even if it exceeds their requirement. As far as sugarcane pricing is related, government administered Statutory Minimum Price (SMP) which acts as a floor. States like UP, Haryana and Punjab fix a higher price for cane, called State Advise Price (SAP), and the SAP has been as high as 20-30 percent above SMP. Government mandates 10 to16 percent of sugar to be sold as levy quota sugar at prices much lower than then market. Government also specifies monthly release quotas for free sale of sugar. 5 As per government regulations, farmers are eligible to sell their crop at SMP. Hence, their profit is impacted by differential between SMP and the cost of cultivation, harvesting and transportation, which are in turn depend on farm productivity, labor availability, distance between farm and market, mode of transport etc., Any price offered to farmers over and above SMP is additional profit to them. The second way of cane marketing is processing the cane into jaggery (gur) and marketing. The processing of sugarcane into jaggery depended on the cane price offered by the sugar mills operating in the area and prices of jaggery. The processors preferred to crush cane into jaggery at times of attractive jaggery prices coupled with low prices offered by sugar mills.

Sugar production in India depends upon sugarcane production. Thus, Sugarcane occupies a prominent position as a cash and commercial crop. Sugarcane is also used for making juice. In the early days, sugarcane was used to produce sugar for the consumption of common people. In India, sugarcane is the third largest crop of the country, in terms of value next to rice and wheat.

In India, the sugar industry is the second largest agro-based industry, next to textile and contributes about Rs.l650 crore to the central exchequer as excise duty and taxes annually. Besides, the state governments realize about Rs.600 crore annually through purchase taxes, cess, etc. The total value of sugarcane produced in the country is estimated at Rs.24000 crore per year.

1.3 DISEASES IN SUGARCANE

Plant disease can be defined as the sum total of abnormal changes in the physiological processes brought about by any biotic or abiotic factor(s) that ultimately threatens the normal

growth and reproduction of a plant which has a great impact on the agricultural yield. Hence, there is a need to identify the plant diseases at a very early stage and take precautionary measures to increase the yield and agricultural productivity.

The most widely used method for plant disease detection is simply naked eye observation by experts through which identification and detection of plant diseases are done. For doing so, a large team of experts as well as continuous monitoring of experts is required, which costs very high when farms are large. At the same time, in some countries, farmers don't have proper facilities or even idea that they can contact to experts. Due to which consulting experts even cost high as well as timeconsuming too. In such a condition, the suggested technique proves to be beneficial in monitoring large fields of crops. And automatic detection of the diseases by just seeing the symptoms on the plant leaves makes it easier as well as cheaper. Plant disease identification by the visual way is a more laborious task and at the same time less accurate and can be done only in limited areas. Whereas if automatic detection technique is used it will take fewer efforts, less time and more accurately. In plants, some general diseases are bacterial, black spotted, and others are Rust, viral and Red cotton Leaf. Image processing is the technique which is used for measuring the affected area of disease, and to determine the difference in the color of the affected are. Image segmentation is the process of separating or grouping an image into different parts. There are currently many different ways of performing image segmentation, ranging from the simple thresholding method to advanced color image segmentation methods. The segmentation process is based on various features found in the image. This might be color information, boundaries or segment of an image. Based on features extracted from the processed image the plant disease is identified and classified.

1.4 OVERVIEW ON PRESENT WORK

Many studies and research have been conducted on plant disease recognition system , a majority of which lack accuracy, and the usage of varied techniques. In [1], B SRAVYA REDDY, R DEEPA, S SHALINI, P BHAGYA DIVYA, in their research, 'Novel Machine Learning Based Approach for Detection and Classification of Sugarcane Plant Disease by Using DWT', disease detection is done using discrete wave length transform algorithm and decision tree approach to classify the disease. Infected plant images are stored in the database and are compared with the input image and classification is done using decision tree. But results are unstable, meaning that a small change in data can lead to large change in the structure of the optimal decision tree. In [2], presented byTISEN HUANG, RUI YANG, WENSHAN HUANG, YIQI HUANG,

XIQIAO. "Detecting Sugarcane Borer Diseases Using Support Vector Machine", Sugarcane borer disease is detected using greyscale conversion for image processing, and SVM algorithm for classification is applicable only for linear dataset. The proposed system is confined to only detect stem related disease.

In [3] by MOSBAH EL SGHAIR, RAKA JOVANOVIC, MILAN TUBA "An Algorithm for Plant Diseases Detection Based on Color Features", In this ,Four different color models were tested and compared: RGB, YCbCr, HSI color model. Median filter is employed for image smoothing (Noise reduction). Kapur's thresholding for plant diseases detection was proposed. But the system faces Disturbance due to vein is present in RGB models. Calculative dimensions of disease spot is not considered.

In [4]S SATHIAMOORTHY, R PONNUSAMY, M NATARAJAN. "Sugarcane Disease Detection Using Data Mining Techniques", the system uses R Datasets are used for experimental studies and Techniques like k means , MLP are used to predict sugarcane leaf disease .But R Dataset considers only limited attributes and thus cannot be generalized.

In [5] TRIMI NEHA TETE, SUSHMA KAMLU "Plant Disease Detection Using Different Algorithms", paper illustrates Two different segmentation techniques: Thresholding and K-means clustering algorithm and classification technique such as Artificial neural network (feed forward back propagation). But thresholding cannot be applied on images with low variation. And it is difficult to predict the k value manually.

PROBLEM STATEMENT

To design and develop a system for detection and prevention of major sugarcane diseases.

1.5 OBJECTIVES

1. Design an efficient system which identifies the disease in sugarcane crop.

2. Use image processing, classify the disease and provide accurate remedy.

3.   Help the farmers to enhance their crop yield by reducing the crop loss due to preventable diseases.

1.6 ORGANIZATION OF REPORT

This report deals with the implementation of the project ''SUGARCANE CROP SUPPORT SYSTEM". This report is organized as 7 chapters, namely, introduction, Literature Review,

System Requirements, System Designs / Methodology, implementation, Results and Discussion and lastly conclusion and future work.

Chapter 1 gives a brief introduction about agriculture in India. It mainly focuses on Sugarcane crop which is the second largest contributor to the agriculture industry.It also gives brief description about various sugarcane crop diseases. It consists of description of background, overview of the present work, problem statement and objectives.

Chapter 2 deals with the detailed analysis of previous studies and research conducted. It consists of summary of prior works, outcome of the review- problems identified, and proposed works.

Chapter 3 specifies the System Requirements. It consists of specification of all functional, and non-functional requirements along with the software/ hardware that has been used in the project.

Chapter 4 specifies the design details. Design is the process of establishing a system that will satisfy the previously identified functional and non- functional requirements. It also consists of specification of the architecture of the system and the major algorithm incorporated.

Chapter 5 includes the implementation part. Implementation is the process of converting the system design into an operational one. This phase starts after the completion of the development phase and must be carefully planned and controlled as it is a key stage. It includes a list of main packages, some of the user- defined functions and some sample code.

Chapter 6 includes the description and the discussion of the results obtained. It consists of Testing and Results. Testing includes the Testing part which is an investigation conducted to provide stakeholders with information of the quality of the product or service under test. It also gives a business an opportunity to understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs.

Chapter 7 mentions the conclusion and future enhancements for the project. IT contains the summary of the work carried out, contributions and utility. Also mentioned are the glossary, acronyms, and the bibliography.

CHAPTER 2

LITERATURE SURVEY

2.1 SUMMARY OF PRIOR WORKS

[1] Describes disease detection is done by using Discrete wavelength transform (DWT) algorithm. There are some set of diseases in the database which check with the already stored input images. Digital camera or similar devices are used to take the pictures and stored in the data set which are different types of leaf images and those are used to identify the affected areas in a leaf. There are different types of techniques used to process those images to get different and useful features needed for the purpose of analysing later.

The methodology discussed in this paper is given below in step by step approach for the proposed image recognition and segmentation process.

1.1 Image Acquisition

In this section, disease affected leaf image is considered as an input image from the dataset of disease affected leaves.

1.2 Image Pre-Processing

After insertion of image, image is pre-processed. Preprocessing is performed to decrease the noise rate and improve the contrast of the image using filters. Spatial filters is a operation where each pixel value is changed by function of intensity of pixel of the neighborhood image.

1.3 Image Segmentation

Enhanced image is segmented using edge detection method. To highlight the affected part and mask the green pixels and compares with the part which is turned in to another color.

1.4 Feature Extraction

Here, diseases affected region of interest is selected from the segmented images. Then, Convert the RGB color (ROI) image into gray scale image and maintain the color Co Occurrence Matrix.

1.5 Classification:

Here, the diseases are classified based on the type of the fungus, bacteria, pathogen or virus the plant is affected.

1.6 Disease detection:

Here the detection of disease is the final step, based on the image the affected part is detected and disease is identified . Obtain the useful segments to classify the leaf diseases. Segment the components using DWT algorithm.

In this paper, approach based on image processing to first detect and then classify leaves according to diseases is used. Here, image acquisition is performed by considering RGB colour disease affected leaf image. Pre-processing of an image is done to enhance the image using filtering. Image segmentation is performed by making use of threshold value. Image feature extraction is performed to obtain the features of leaf disease symptoms. Image classification is performed using Decision tree (DT).

[2] This paper addresses a specific disease called Stem Borer disease that infects a sugarcane plant, where the larvae feeds on the stem. From the images, the sugarcane borer diseases are characterized by different sizes of approximate elliptic wormhole and theirs color are black.

Considering that there was a sag in the wormhole, which was not sensitive to the light reflection, so the grey value of wormhole is lower and closer to black, while other parts are higher and closer to white or cinereous because of the high exposure.

Preprocessing of image is done as follows:

1) The target image and part area of the target were obtained by segmentation which the threshold was 150, and the connected domain was divided into disconnected area, then the largest area which was selected by area method was the target area of sugarcane seed. This step mainly reduced the influence of the background on the segmentation result and achieved the sugarcane target region which includes some noises and the wormhole target.

2) Now select the wormhole. Removed the region which area of 1 and selected the region corresponding to the minimum average grey value for the wormhole after calculating the area and average grey value of each non-connected region. As a result, the region corresponding to the minimum average grey value was the result that we segmented.

Classification is done using SVM.

The selection of the SVM has been carried out taking into account the computational resources was requiredby it for making a decision. But, the SVM approach has a good classification effect.

The minimum average grey value and the corresponding minimum grey value can effectively avoid miscalculation of pseudo diseases for the wax and leaf scar. Compared to RBF kernel function and Polynomial kernel function, the recognition rate is more stable and reliable when RBF kernel function as inner product function. This shows that the RBF kernel function can be used as the kernel function of the support vector machine to solve the problem of sugarcane borer disease detection.
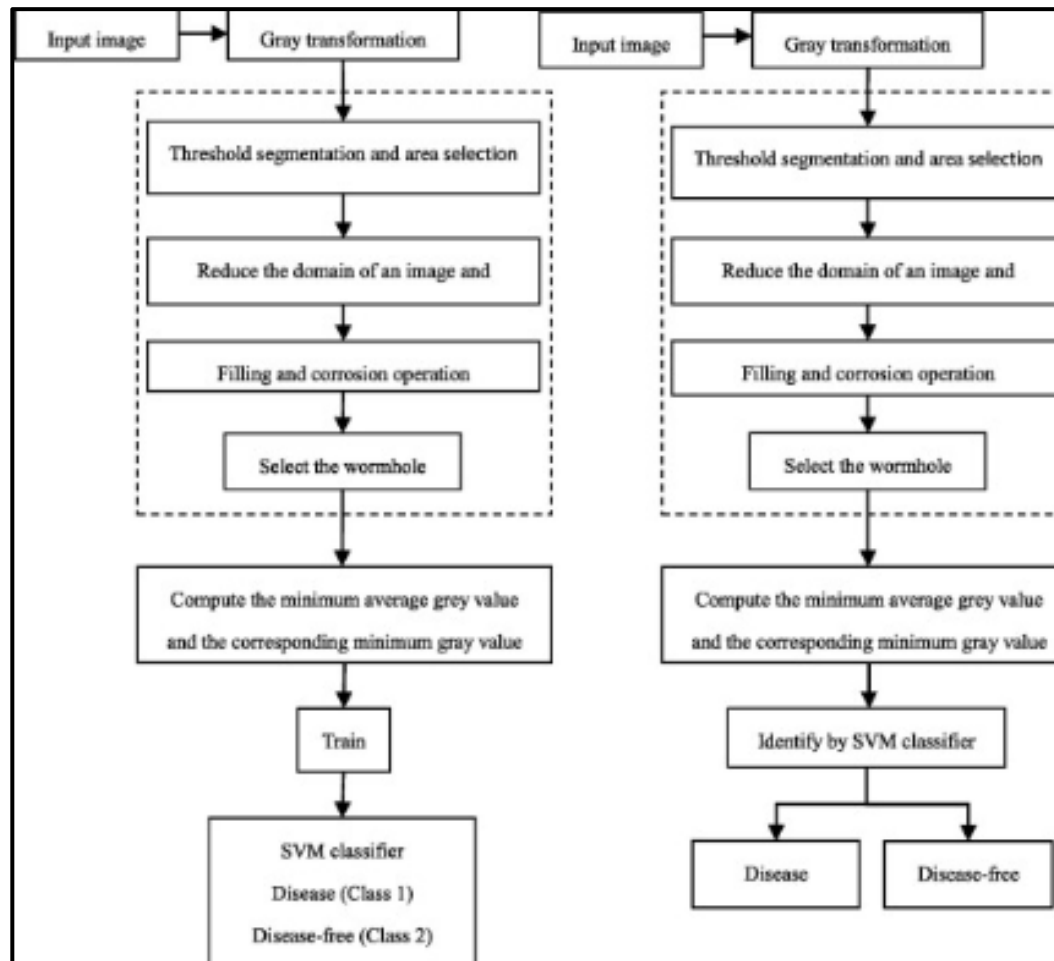

Fig 2.1: Flowchart followed

[3] An Algorithm for Plant Diseases Detection Based on Color Features.ByMosbah El Sghair, RakaJovanovic, Milan Tuba published at IJAS Vol.2 in 2017.

In this paper four different color models were tested and compared: RGB, YCbCr, HSI color model. Median filter is employed for image smoothing (Noise reduction). Kapur'sthresholding for plant diseases detection was proposed.

In plants, leaf vein is totally different in intensity and disease spot is different in color compared to plant leaf. Therefore if Kapur's method is applied on grayscale image, vein will be

detected in binary image with the disease spot. However the region of interest is simply disease spots, not vein. For minimize the effect of presence of vein, RGB color model is not suitable for segmentation. Thresholding method are often applied on color element to discover disease spot accurately.

The next step in proposed algorithm is image smoothing. During image assortment, some noise is also introduced due to camera flash. This noise might have an effect on the detection of disease. To remove unneeded spots, image smoothing technique is required. In this paper adjusted median filter is employed for this purpose.

After image smoothing, a method to detect and isolate thedisease spot is required. It is necessary to find a threshold value that will differentiate the disease spots from plant leaf. One of the most used method for thresholding is Kapur's method that is based on the entropy. This method maximize the amount of information between the two parts of a intensity histogram that are separated by concrete threshold value or better to say maximize the entropy measure of the part of the histogram in order to each part has a more centralized distribution.

In this paper a method based on different color modelsand Kapur'sthresholding for plant diseases detectionwas proposed. Four different color models were testedand compared: RGB, YCbCr, HSI and CIELAB color model. The best results were obtained when HSI color model was used. Component H was used for image segmentation where diseases were separated from the leaf. Median filter was applied to color transformed image. At the end, disease spots area are determined

by applying Kapur's threshold on different color components. Experimental result shows that noise that is introduced due to background, vein and camera flash makes the least problem for HSI color model. Following this technique totally different disease spots are detected accurately and results do not seem to be laid low with background, sort of leaf, type of disease spot and camera.

[4]Sugarcane Disease Detection Using Data Mining Techniques. Published by S Sathiamoorthy, R Ponnusamy, M Natarajan in the year 2018

In the paper, the R-Dataset is used as a benchmark database to perform experimental study to find out the diseases which affects the sugarcane leaf. Classification algorithms like J48, pruned tree and Multilayer perceptron were analysed and compared with K-means clustering algorithm. These algorithms were implemented in WEKA tool forclassification and clustering.

J48 pruned trees algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm the critical distribution of the data is easily understandable. The WEKA tool provides a number of options associated with tree pruning. In case of potential over fitting pruning can be used as a tool for précising. In other algorithms the classification is performed recursively till every single leaf is pure, that is the classification of the data should be as perfect as possible. This algorithm it generates the rules from which particular identity of that data is generated. The objective is progressively generalization of a decision tree until it gains equilibrium of flexibility and accuracy.

Basic Steps in the Algorithm:

i. In case the instances belong to the same class the tree represents a leaf so the leaf is returned by labelling with the same class.

ii. The potential information is calculated for every attribute, given by a test on the attribute. Then the gain in information is calculated that would result from a test on

the attribute.

iii. Then the best attribute is found on the basis of the present selection criterion and that attribute selected for branching.

[5] An Identification Of Crop Disease Using Image Segmentation published by K. Vinoth Kumar and T. Jayasankar in the year 2018. In this paper the methodology is broken down into two segments:

(a) Image Processing Segment: Where the properties of the leaf image will be enhanced segmented from the background.

(b) Pattern Recognition Segment: Where the required features will be extracted and this information will be matched with the predefined knowledge about the plant diseases for detecting which disease has actually affected the plant.

Pattern recognition further includes:

Feature Extraction: It is the procedure of outlining a set of necessary features, or image characteristics that form the core element which when represented in an efficient or meaningful manner give the required information that is important for analysis and classification purpose.

Disease Identification: It is the process of understanding the meaning of the feature extracted from the image and matching the extracted information with the predefined set of rules and thus coming to a conclusion. The result of the computation is obtained in this step:

Addressing a fundamental problem in programmable matter, The first deterministic algorithm to elect a unique leader in a system of connected amoebots assuming only that amoebots are initially contracted. Previous algorithms either used randomization, made various assumptions (shapes with no holes, or known shared chirality), or elected several co-leaders in some cases. Some of the building blocks we introduce in constructing the algorithm are of interest by themselves, especially the procedure we present for reaching common chirality among the amoebots as addressed in "Deterministic Leader Election in Programmable Matter(2019)" by Yuval Emek, Shay Kutten, Ron Lavi and William K. Given the leader election and the chirality agreement building block, it is known that various tasks in programmable matter can be performed or improved.

When considering recently proposed and realized systems of programmable matter, one can distinguish between passive and active systems. In passive systems, computational units cannot control their movements and have (atmost) very limited computational abilities, relying instead on their physical structure and interactions with the environment to achieve locomotion (e.g., Woods 2015; Angluin et al. 2006; Reid and Latty 2016). A large body of research in molecular self-assembly falls under this category, which has mainly focused on shape formation (e.g., Douglas et al. 2009; Cheung et al. 2011; Wei et al. 2012). In contrast, the work examines building dynamic bridges whose exact shape is not predetermined. Mohammed et al. studied a similar problem of connecting DNA origami landmarks with DNA nanotubes, using a carefully designed process of nanotube nucleation, growth, and diffusion to achieve and maintain the desired connections (Mohammed et al. 2017). Significant differences between their approach and ours are: (1) the bridges considering already connect their endpoints at the start and focus on the specific goal of optimizing their shape with respect to a parameterized objective function, and (2) the system is active as opposed to passive.

Addressing a fundamental problem in programmable matter, a presentation on the first deterministic algorithm to elect a unique leader in a system of connected amoebots assuming only that amoebots are initially contracted. Previous algorithms either used randomization, made various assumptions (shapes with no holes, or known shared chirality), or elected several co-leaders in some cases. Some of the building blocks we introduce in constructing the algorithm are of interest by themselves, especially the procedure we present for reaching common chirality among the amoebots. Given the leader election and the chirality agreement building block, it is known that various tasks in programmable matter can be performed or improved. The main idea of

the new algorithm is the usage of the ability of the amoebots to move, which previous leader election algorithms have not used.

Active systems are composed of computational units that can control their actions to solve a specific task. Examples include swarm robotics, various other models of modular robotics, and the amoebot model, which is the computational framework (detailed in Sect. 1.2).

Swarm robotic systems usually involve collections of autonomous robots moving freely in space with limited sensing and communication ranges. These systems can perform a variety of tasks including gathering (Cieliebak et al. 2012), shape formation (Flocchini et al. 2008;Rubenstein et al. 2014), and imitating the collective behaviour of natural systems (Chazelle 2009); however, the individual robots typically have more powerful communication and processing capabilities than those considered.

Modular self-reconfigurable robotic systems focus on the motion planning and control of kinematic robots to achieve dynamic morphology (Yim et al. 2007), and metamorphic robots form a subclass of self-reconfiguring robots (Chirikjian 1994) that share some characteristics with the geometric amoebot model. Walter et al. have conducted some algorithmic research on these systems (e.g., Walter et al. 2004a, b), but focus on problems disjoint from those considered.

In the context of molecular programming, the model most closely relates to the nubot model by Woods et al.(2013), Chen et al. (2015), which seeks to provide a framework for rigorous algorithmic research on self-assembly systems composed of active molecular components, emphasizing the interactions between molecular structure and active dynamics. This model shares many characteristics with the amoebot model (e.g., space is modelled as the triangular lattice, nubot monomers have limited computational abilities, and there is no global orientation) but differs in that nubot monomers can replicate or die and can perform coordinated rigid body movements. These additional capabilities prohibit direct translation of results under the nubot model to the amoebot model.

In the article Programming Modular Robots with Locally Distributed Predicates authored by Michael De Rosa, Seth Copen Goldstein, Peter Lee, Jason D. Campbell, and Padmanabhan Pilla, a description on the hardware and software challenges involved in realizing Claytronics, a form of programmable matter made out of very large numbers-potentially millions-of submillimeter sized spherical robots. The goal of the claytronics project is to create ensembles of cooperating submillimetre robots, which work together to form dynamic 3D physical objects. For example, claytronics might be used in telepresence to mimic, with high-fidelity and in 3-

dimensional solid form, the look, feel, and motion of the person at the other   end of the telephone call. To achieve this long-range vision, also investigating hardware mechanisms for constructing submillimetre robots, which can be manufactured en masse using photolithography. The proposal of the creation of a new media type, which is called pario. The idea behind pario is to render arbitrary moving, physical 3-dimensional objects that you can see, touch, and even hold in your hands. In parallel with the hardware effort, a novel distributed programming languages and algorithms to control the ensembles, LDP and Meld. Pario may fundamentally change how to communicate with others and   interact with the world around us. The research results to date suggest that there is a viable path to implementing both the hardware and software necessary for claytronics, which is a form of programmable matter that can be used to implement pario. While a significant progress is made, there is still much research ahead in order to turn this vision into reality.

A novel shape formation algorithm for ensembles of 2-dimensional lattice-arrayed modular robots, based on the manipulation of regularly shaped voids within the lattice (``holes"). The algorithm is massively parallel and fully distributed. Constructing a goal shape requires time proportional only to the complexity of the desired target geometry. Construction of the shape by the modules requires no global communication nor broadcast floods after distribution of the target shape. Results in simulation show 97.3\% shape compliance in ensembles of approximately 60,000 modules, and believe that the algorithm will generalize to 3D and scale to handle millions of modules as explained in Scalable by Michael De Rosa, Seth Copen Goldstein, Peter Lee, Jason D. Campbell and Padmanabhan Pillai, 2006 IEEE International Conference.

Programmable matter refers to a technology that will allow one to control and manipulate three-dimensional physical artifacts (similar to how already control and manipulation of two-dimensional images with computer graphics) as explained in Claytronics: An Instance Of Programmable Matter, Seth Copen Goldstein and Todd C. Mowry. In other words, programmable matter will allow us to take a (big) step beyond virtual reality, to synthetic reality, an environment in which all the objects in a user's environment (including the ones inserted by the computer) are physically realized. Note that the idea is not to transport objects nor is it to recreate an objects chemical composition, but rather to create a physical artifact that will mimic the shape, movement, visual appearance, sound, and tactile qualities of the original object.
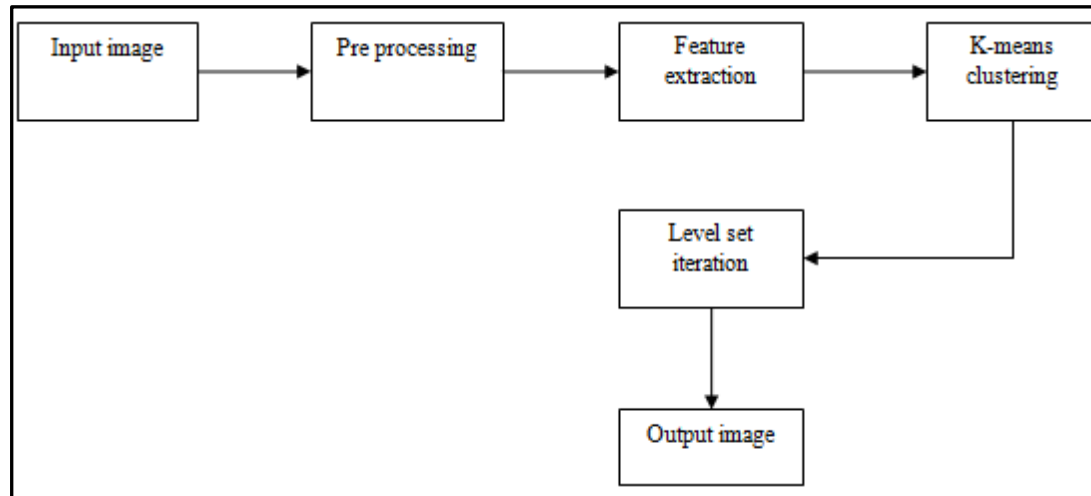
Fig 2.2: Flowchart followed

2.2 OUTCOME OF THE REVIEW-PROBLEMS IDENTIFIED

There are many methods in automated or computer vision plant disease detection and classification process, but still, this research field is lacking. In addition, there are still no commercial solutions on the market, which are easily accessible by farmers to recognize the diseases affecting their yield.

In the above discussed papers various algorithms like SVM and decision tree have been used. SVM can give a greater accuracy for classification only when the dataset is linear. Classification done using decision tree yield unstable results even a small change in data can lead to large change in the structure of the optimal decision tree. R data set has been considered to train the model for classification,which addresses the disease that are specific to only certain region on the globe(Brazilian region)and cannot be used to identify the disease globally.

2.3 PROPOSED WORK

Considering the performance issues as mentioned above, and the lack of representative systems, a system that is capable of achieving a higher performance in classifying the 4 sugarcane diseases eye spot, red rot, wheat rust, yellow leaf, is developed employing several machine learning algorithms along with the associated analysis.

The dataset is created by clicking the images of the diseased plants from the fields. With limited dataset   the prediction for disease classification will not give higher accuracy, so to overcome this issue and increase the size of dataset image augmentation technique is deployed. The images in the dataset are then pre-processed to resize and remove noise.

The Pre-processed images are then fed into the pre-trained MobileNetV2 model which is retrained according to the new customized dataset to identify and classify the sugarcane disease into the four major diseases addressed. The Trained model is then obtained as the saved model which is further fed into the TensorFlowLite convertor to obtain .tflite file which can then be deployed in android applications.

New disease suspected plant images are then clicked and uploaded to the application where the trained model executes to identify if the plant is diseased and classified accordingly. The application also helps farmers by suggesting the remedial measures to be taken to address the particular disease at the early stage thereby decreasing the crop loss.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

Requirement analysis is very critical process that enables the success of a system or software project to be assessed which are split into two types: Functional and Non-Functional requirements. Functional Requirements define the internal working of the software, i.e., the calculations, technical details, data manipulation and processing and other specific functionalities that show how the cases are to be satisfied and how they are supported by non-functional requirements, which impose constraints on the design or the implementation. Behavioural requirements describe all the cases where the system uses the functional requirements; these are captured in use cases. It defines a system's reaction to particular inputs at a component level.

Mobile devices which can also be called handheld devices like mobile phones, tablets, personal digital assistants (PDAs), or enterprise digital assistants need mobile applications for optimal usefulness and service delivery. Mobile applications are to be taken as utility software packages that enhance full functionality and the mobile devices' ability to meet diverse users' needs. The purpose of a well-done requirement analysis phase is to archive a user-friendly system by being a user, to acknowledge the appropriateness of the takes.

The following are the Functional requirements of our system:

1. Image acquisition for dataset preparation.
2. Image augmentation and pre-processing.
3. Training the model for disease identification and classification.
4. Deployment of trained model into android application.
5. Identification and Classification of diseased plants with suggested remedies.

3.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement places constraint on "How should the software system fulfil the functional requirements?". It defines the system's properties and constraints applied to it as awhole. It describes system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

The following are the Functional requirements of our system:

1. Performance

2. Dependability

3. Capacity

4. Operability

5. Interoperability

3.3 SOFTWARE/HARDWARE USED

HARDWARE REQUIREMENTS

- Android Version : Android 7 and above

- Processor   : Intel Atom, 1.2 Ghz, more faster processors

- Storage   : 12 GB and above

- RAM   : 4 GB and above

SOFTWARE REQUIREMENTS

- Language used : JAVA, Python and XML

- Frameworks   : Tensorflow 2.0, Mobilenet V2

- Jupyter Notebook

- Android Studio

3.4 LANGUAGE/ FRAMEWORK USED

3.4.1 PYTHON

History of Python

Python is an interpreter, high-level, general-purpose programming language created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, introduced features like list comprehensions and a garbage collection system

capable of collecting reference cycles. Python 3.0, released 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages Python and CPython.

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible Python 2.6 and 2.7.

In February 1991, van Rossum published the code (labelled version 0.9.0) to alt.sources. Already present at this stage in development were classes with inheritance, exception handling, functions, and the core datatypes of list, dict, str and so on. Also in this initial release was a module system borrowed from Modula-3; Van Rossum describes the module as "one of Python's major programming units". Python's exception model also resembles Modula-3's, with the addition of an else clause. In 1994 comp.lang.python, the primary discussion forum for Python, was formed, marking a milestone in the growth of Python's userbase.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

The latest version of Python is 3.8.3; however the latest stable release of Python is version 3.8.2, released on 24 February 2020.

Python Programming Language

In the python programming language, all source code is first written in plain text files ending with the .py extension.

Python does not convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code.

The Python interpreter performs following tasks to execute a Python program:

- **Step 1:**The interpreter reads a python code or instruction. Then it verifies that the instruction is well formatted, i.e. it checks the syntax of each line.If it encounters any error, it immediately halts the translation and shows an error message.

- **Step 2:** If there is no error, i.e. if the python instruction or code is well formatted then the interpreter translates it into its equivalent form in intermediate language called "Byte code".Thus, after successful execution of Python script or code, it is completely translated into Byte code.

- **Step 3:** Byte code is sent to the Python Virtual Machine(PVM).Here again the byte code is executed on PVM.If an error occurs during this execution then the execution is halted with an error message.
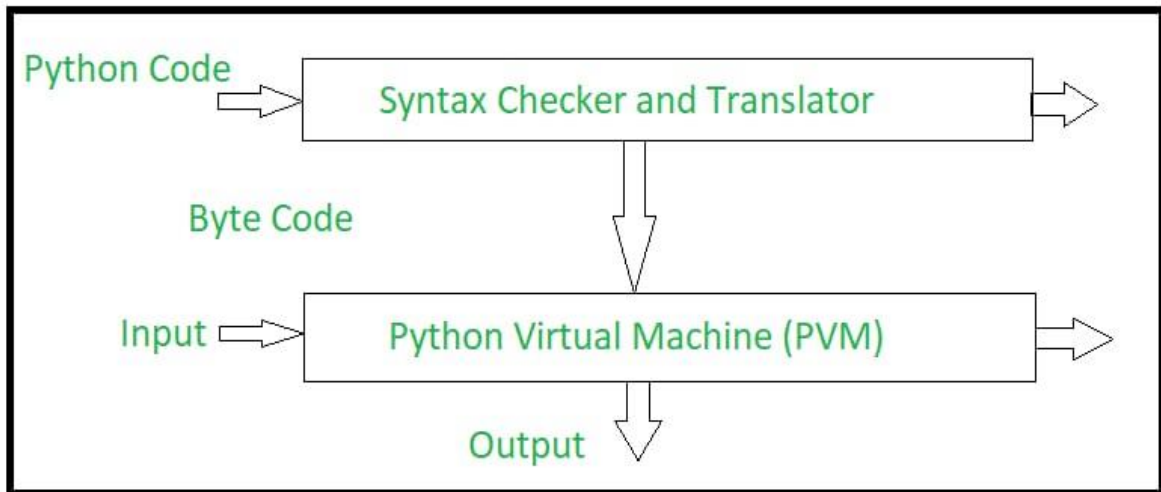
Fig 3.1: An overview of Python Program Conversion

3.4.2 TENSORFLOW 2.0

TensorFlow started as an open-source deep learning library and has today evolved into an end to end machine learning platform that includes tools, libraries and resources for the research community to push the state of the art in deep learning and developers in the industry to build ML & DL powered applications.

TensorFlow had its first public release back in 2015 by the Google Brain team. At the time, the evolving deep learning landscape for developers & researchers was occupied by Caffe and Theano. In a short time, TensorFlow emerged as the most popular library for deep learning.

TensorFlow is fast with backend written in C++ and has interfaces in Python, Java, Swift, and Android.

TensorFlow 2.0 makes development of ML applications much easier. With tight integration of Keras into TensorFlow, eager execution by default, and Pythonic function execution.

The standardized SavedModel file format can be used to run models on a variety of runtimes, including the cloud, web, browser, Node.js, mobile and embedded systems.
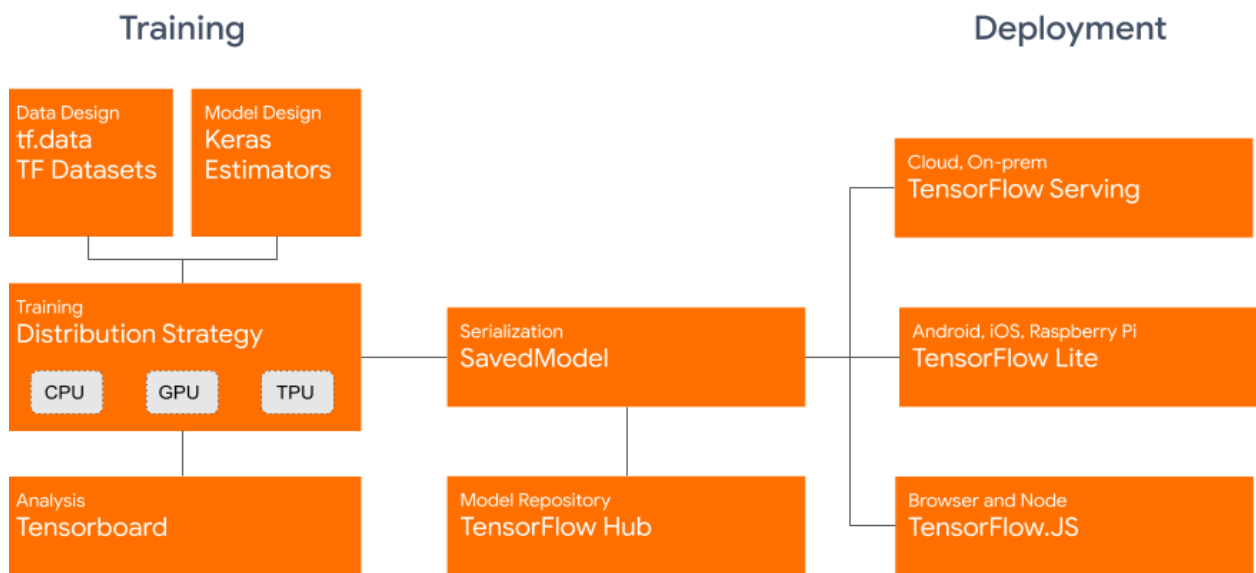

Fig 3.2: Tensorflow 2.0 Architecture

3.4.3 KERAS

tf.keras is TensorFlow's implementation of the Keras API specification. This is a high-level API to build and train models that includes first-class support for TensorFlow-specific functionality, such as eager execution, data pipelines, and Estimators. keras makes TensorFlow easier to use without sacrificing flexibility and performance.

In Keras, you assemble *layers* to build *models*. A model is (usually) a graph of layers. The most common type of model is a stack of layers: the `tf.keras.Sequential` model.

3.4.4 TENSORFLOW LITE

TensorFlowLite is a set of tools to help developers run TensorFlow models on mobile, embedded, and IoT devices. It enables on-device machine learning inference with low latency and a small binary size.

TensorFlowLite consists of two main components:

The **TensorFlowLiteinterpreter**, which runs specially optimized models on many different hardware types, including mobile phones, embedded Linux devices, and microcontrollers.

The **TensorFlowLiteconverter**, which converts TensorFlow models into an efficient form for use by the interpreter, and can introduce optimizations to improve binary size and performance.

TensorFlowLite is designed to make it easy to perform machine learning on devices, "at the edge" of the network, instead of sending data back and forth from a server. For developers, performing machine learning on-device can help improve:

Latency: there's no round-trip to a server

Privacy: no data needs to leave the device

Connectivity: an Internet connection isn't required

Power consumption: network connections are power hungry

TensorFlowLite works with a huge range of devices, from tiny microcontrollers to powerful mobile phones.

Key features

- *Interpreter tuned for on-device ML*, supporting a set of core operators that are optimized for on-device applications, and with a small binary size.
- *Diverse platform support*, covering Android and iOS devices, embedded Linux, and microcontrollers, making use of platform APIs for accelerated inference.
- *APIs for multiple languages* including Java, Swift, Objective-C, C++, and Python.
- *High performance*, with hardware acceleration on supported devices, device-optimized kernels, and pre-fused activations and biases.
- *Model optimization tools*, including quantization, that can reduce size and increase performance of models without sacrificing accuracy.
- *Efficient model format*, using a FlatBuffer that is optimized for small size and portability.
- *Pre-trained* models for common machine learning tasks that can be customized to your application.
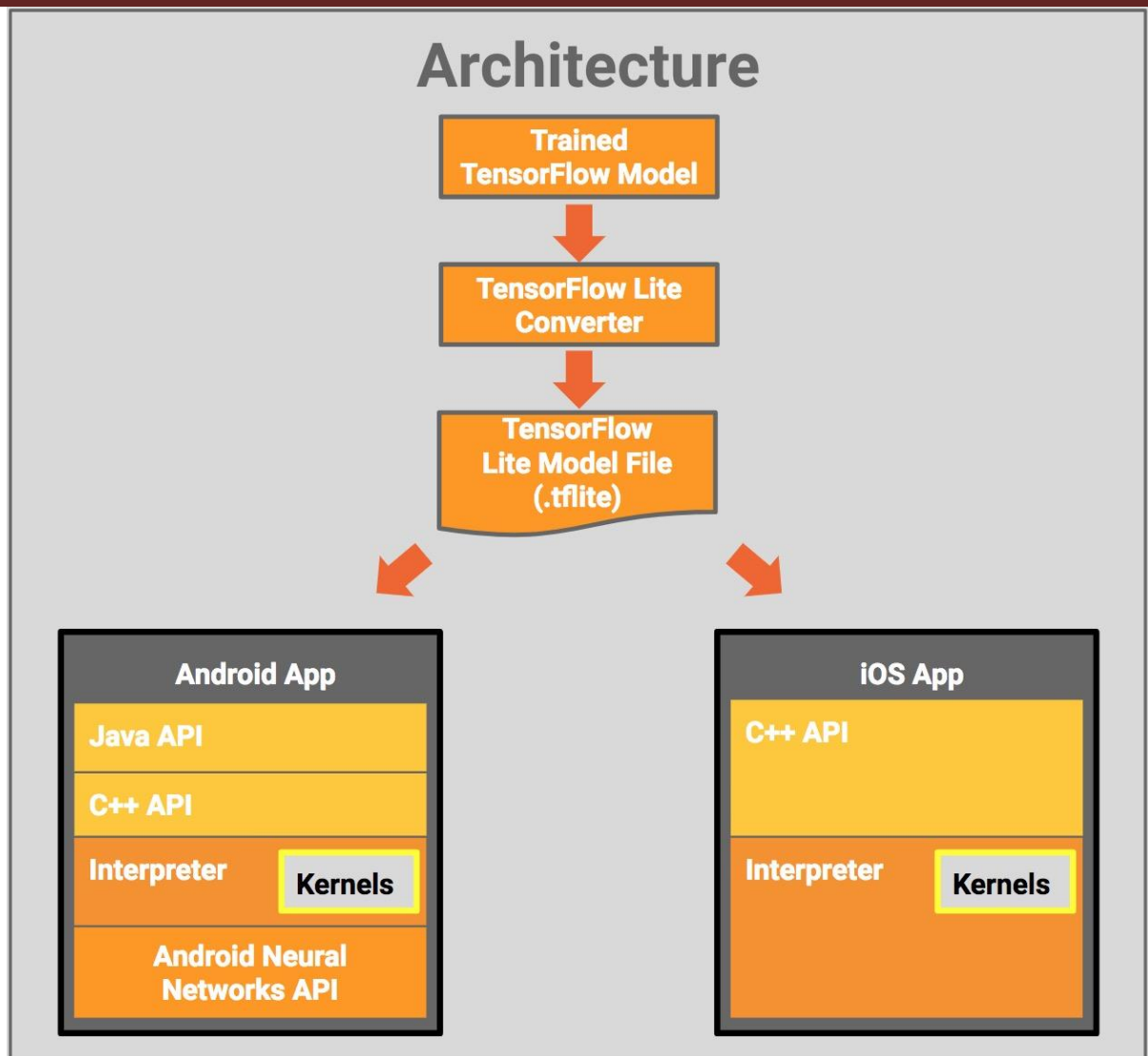
Architecture

Fig 3.3: TensorflowLite Architecture

3.4.5 ANDROID

Android or Android technology or more emphatically Android operating platform is a stack of software for mobile devices. This stack of mobile devices consists of different layers that include an operating system, middleware and key applications to run on the operating platform.

Android Application Framework

Now like most of the major software and operating platforms on the earth Android also comes with a software development kit which is termed commonly as Android SDK. Android SDK provides you the API libraries and tools for building and developing new applications on Android operating environment using the java programming language. This procedure of

developing the applications on Android platform in java programming language using the tools and API libraries provided by Android SDK is called as Android Application Framework.

Basic Features supported Android Application Framework.

Android Application Framework supports the features that made us use and enjoy the wide range of applications for variety of uses. Here are some of the important features:

- WebKit engine based integrated browser.

- Optimized graphics powered by the advanced graphics library.

- SQL for storage of structured data.

- For various types of video, audio and image formats media support.

- Device emulator, tools for debugging, etc.

In the above mentioned list we did not mention some of the hardware dependant features as these tend to largely vary as per the device, though nevertheless android application framework support them. Some of the device dependant features supported by android include GSM telephony, network connection profiles such as Bluetooth, Edge, 3G, WiFi, utility features such as camera, compass, GPS, etc.
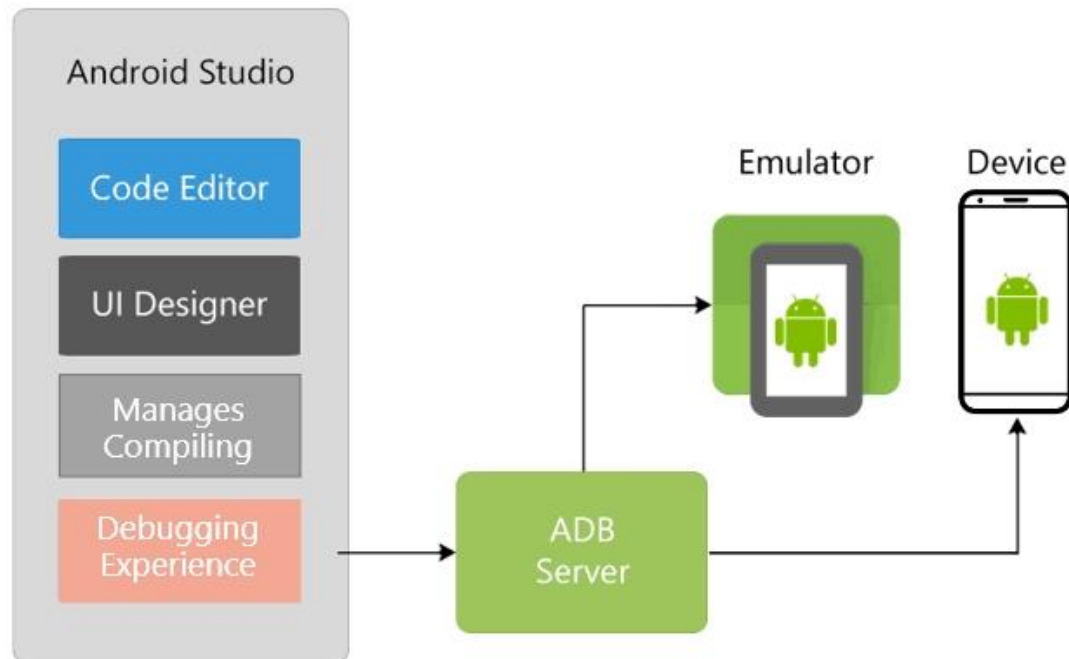
Fig 3.4: Android Application Architecture

To explore emergent phenomena that result from collections of entities with limited mobility and sensing, developed what it is called as "smarticles". Smarticles, or smart particles, are

small 14 × 2.5 × 3 cm robots which can change their shape in situ, but are incapable of rotating or displacing individually. Each smarticle is a three-link, two revolute joint, planar robot where only the center link is in contact with the ground.

Each smarticle consists of two Power HD-1440A MicroServos, a MEMS analog omnidirectional microphone, two photoresistors, a current sensing resistor, and a re-programmable Arduino Pro Mini 328–3.3V/8MHz, which handles the ADC and servo control. The two servos control the smarticle's two outer links, allowing the smarticle to fully explore its two-dimensional configuration space. The microphone and pair of photoresistors represent two channels through which can send basic commands: using varying frequency ranges of sound or controlling levels of light. The current sensing resistor detects current draw from the servos, and thus the torque they are experiencing, allowing each smarticle to sense its own stress state. The links of the smarticles were 3D printed, ensuring uniform construction between all smarticles. Each smarticle is capable of performing predefined shape changes in the joint space. When placed, a collection of smarticles inside an unanchored ring, calling this new assemblage a supersmarticle.

3.4.5 JAVA FOR ANDROID

Java is the official language of Android development and is supported by Android Studio.Java itself was released by Sun Microsystems back in 1995 and is used for a wide range of programming applications. Java code is run by a "virtual machine," which runs on Android devices and interprets the code.

Fig 3.5: JAVA for Android

The following set of APIs are supported when building your app using Android Gradle plugin 4.0.0 or higher:

Sequential streams (java.util.stream)

- A subset of java.time

- java.util.function

- Recent additions to java.util.{Map,Collection,Comparator}

- Optionals (java.util.Optional, java.util.OptionalInt and java.util.OptionalDouble) and some other new classes useful with the above APIs

- Some additions to java.util.concurrent.atomic (new methods on AtomicInteger, AtomicLong and AtomicReference)

- ConcurrentHashMap (with bug fixes for Android 5.0)
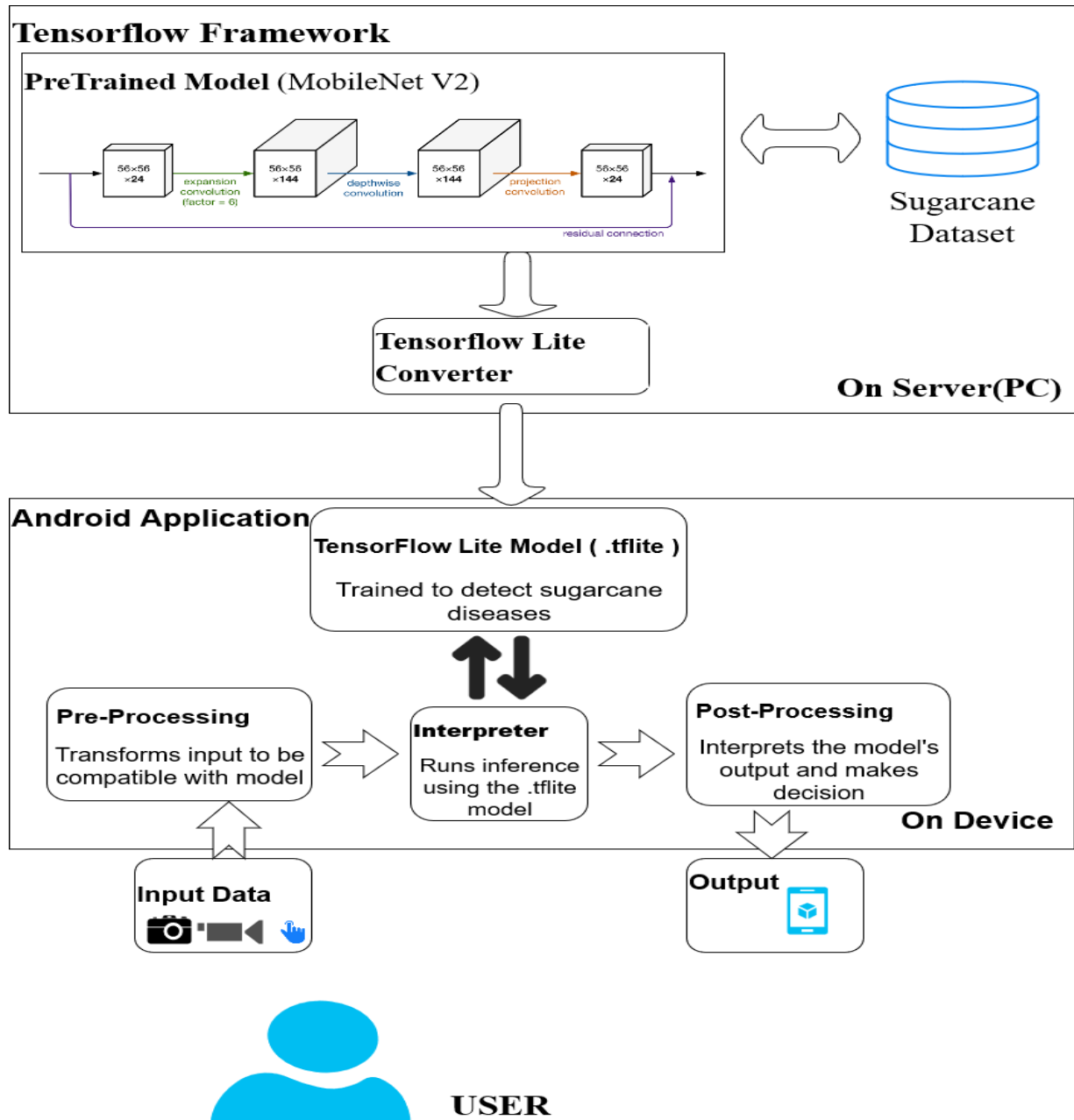
CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURE

Fig 4.1: System Architecture of Proposed System

The components of the system architecture are explained below:

- **USER:** The user here is the client who would be using the android application. The user needs to upload an image of the region to be inspected for diseases in a sugarcane crop. On which the inference will be returned to the user on device.

- **Android Application:** It consists of the TensorFlowLite model which will perform on device inference by invoking the TensorFlowLite Interpreter. It also pre-processes the input data to be compatible with the .tflite model and post-processes the output result and displays the final Result to the user.

- **TensorFlow Framework:** A pre-trained model(here, **MobileNet V2)** is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. This pre-trained model is the base model on which the Sugarcane dataset is trained to obtain a custom model. This is a resource intensive process that occurs in the Server machine and is known as Transfer Learning. The Custom CNN model is saved and converted to .tflite format by TensorFlowLite Converter.

4.2 MAJOR ALGORITHM

MobileNetV2 model

MobileNetV2 is a neural network model developed at Google, and pre-trained on the ImageNet dataset, a large dataset of 1.4M images and 1000 classes of web images.MobileNets are small, low-latency, low-power models parameterised to meet the resource constraints of a variety of use cases that is optimised for mobile devices.

MobileNetV2 improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. It is a very effective feature extractor for object detection and segmentation. For instance, for detection, when paired with Single Shot Detector Lite, MobileNetV2 is about 35 percent faster with the same accuracy than MobileNetV1.

It builds upon the idea of using depth-wise separable convolutions as efficient building blocks.MobileNet has two new features:

- Linear bottlenecks between the layers: Experimental evidence suggests that using linear layers is crucial as it prevents nonlinearities from destroying too much information. Using non-linear layers in bottlenecks indeed hurts the performance by several percent, further validating our hypothesis

- Shortcut connections between the bottlenecks

THE BASIC STRUCTURE OF MobileNetV2

The bottlenecks of the MobileNetV2 encode the intermediate inputs and outputs while the inner layer encapsulates the model's ability to transform from lower-level concepts such as pixels to higher level descriptors such as image categories. With traditional residual connections, shortcuts enable faster training and better accuracy.
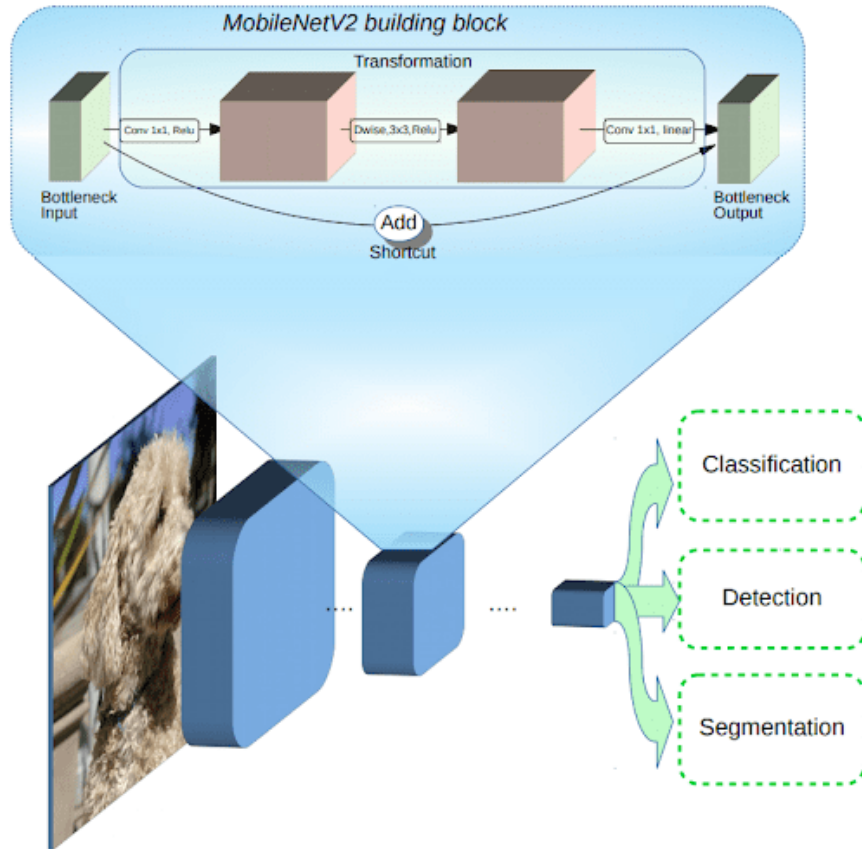
Overview of MobileNetV2 Architecture. Blue blocks represent composite convolutional building blocks as shown above.

Fig 4.2: MobileNetV2 Architecture

The basic building block is a bottleneck depth-separable convolution with residuals. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. The researchers have tailored the architecture to different performance points, by using the input image resolution and width multiplier as tunable hyperparameters, that can be adjusted depending on desired accuracy or performance trade-offs. The primary network (width multiplier 1, 224×224), has a computational cost of 300 million

multiply-adds and uses 3.4 million parameters. The network computational cost ranges from 7 multiply-adds to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters.

MobileNet has several properties that make it suitable for mobile applications and allows very memory-efficient inference and utilises standard operations present in all neural frameworks.Thus MobileNetV2 provides a very efficient mobile-oriented model that can be used as a base for many visual recognition tasks like disease detection in plants.

4.3 LANGUAGE DESCRIPTION

4.3.1 CORE TOOLS AND TECHNOLOGIES

This section covers the complete development matrix. It identifies the complete technologies elements with guidelines and specifications for specific implementation.

Python includes development tools that help to implement the algorithm efficiently. These include the following:

Code Analyzer

Checks the code for problems and recommends modifications to maximize performance and maintainability.

Language Specification

The Python language supports the vector and matrix operation functional to engineering and scientific problems. It enables fast development and execution.

CHAPTER 5

IMPLEMENTATION

5.1 MODULES EXPLAINED

Module 1: Image Acquisition

Appropriate datasets are required at all stages of image recognition research, starting from training phase to evaluating the performance of recognition algorithms. All the images collected for the dataset were manually clicked from the fields.Images in the dataset were grouped into four different classes addressing four major sugarcane diseases *Eye spot, Red rot,White rust* and *Yellow leaf*.

In order to distinguish healthy leaves from diseased ones, one more class was added in the dataset. It contains only images of healthy leaves. An extra class in the dataset with background images was beneficial to get more accurate classification. Thus, deep neural network could be trained to differentiate the leaves from the surrounding.

Next step was to enrich the dataset with augmented images. The main goal of the presented study is to train the network to learn the features that distinguish one class from the others. Therefore, when using more augmented images, the chance for the network to learn the appropriate features has been increased.

Module 2: Image Augmentation and Pre-processing

The main purpose of applying augmentation is to increase the dataset and introduce slight distortion to the images which helps in reducing overfitting during the training stage. In machine learning, as well as in statistics, overfitting appears when a statistical model describes random noise or error rather than underlying relationship . The image augmentation contained one of several transformation techniques including flipping,rotating etc. Functions of keras like ImageDataGenerator(), fit(), keras.preprocessing are used to generate and export augmented image files to a folder in order to build up a giant dataset of altered images.

Furthermore, procedure of image preprocessing involved cropping of all the images manually, making the square around the leaves, in order to highlight the region of interest. During the phase of collecting the images for the dataset, images with smaller resolution and dimension less than 500 px were not considered as valid images for the dataset. In addition, only the images

where the region of interest was in higher resolution were marked as eligible candidates for the dataset. In that way, it was ensured that images contain all the needed information for feature learning. Images used for the dataset were image resized to 224 X 224 to standardize the dataset images.

Module 3: Training the model for disease identification and classification

Training the deep convolutional neural network for making an image classification model from the dataset described. There are several well-known state-of-the-art deep learning based pre-trained models ,of which MobilenetV2 is suitable for both research experiments and industry deployment**.**The pre-processed images were then fed into the MobileNetV2 which is pre-trained on the ImageNet dataset, a large dataset of 1.4M images and 1000 classes of web images.

MobileNetV2 architecture is considered a starting point, but modified and adjusted to support our 4 categories of diseases.

Sophisticated deep learning models have millions of parameters (weights) and training them from scratch often requires large amounts of data of computing resources. Transfer learning is a technique that shortcuts much of this by taking a piece of a model that has already been trained on a related task and reusing it in a new model.

For the Sugarcane disease classification model, the intermediate layer of MobileNetV2 will be used for feature extraction. A common practice is to use the output of the very last layer before the flatten operation, the so-called "bottleneck layer". The reasoning here is that the following fully-connected layers will be too specialized to the task the network was trained on, and thus the features learned by these layers won't be very useful for a new task. The bottleneck features, however, retain much generality.

Thus, an instantiated MobileNetV2 model pre-loaded with weights trained on ImageNet. By specifying the include top=False argument, we load a network that doesn't include the classification layers at the top, which is ideal for feature extraction. A classification head is added on top of this base model (MobileNetV2) with 4 layers for classification.

- **A convolutional layer (Conv2D)** that extracts features from the image or parts of an image.

- Two **subsampling or pooling layer** that reduces the dimensionality of each feature to focus on the most important elements (Dropout and GlobalAveragePooling2D Layers).

- And finally a **fully connected Dense layer** that takes a flattened form of the features identified in the previous layers, and uses them to make a prediction about the image.

Finally to increase the performance even further, train (or "fine-tune") the weights of the top layers of the pre-trained model alongside the training of the classifier layer added. The training process will force the weights to be tuned from generic features maps to features associated specifically to Sugarcane dataset.The trained model is then obtained as the saved model.

Module 4: Deployment of trained model into Android application.

The Trained model which is obtained in form of saved model is not compatible to integrate with Android Application. Hence, TensorflowLite framework is used to convert the saved model into .tflite format which can be deployed into the android Application.

TensorFlowLite has two main components. TensorflowLite convertor and TensorflowLite interpreter. The **TensorFlowLiteconverter** , converts TensorFlow models(the saved model) into an efficient form(.tflite) for use by the interpreter, and can introduce optimizations to improve binary size and performance. The **TensorFlowLite interpreter** then runs specially optimized models on mobile phones to identify and classify the images into appropriate category of disease based on the pre-trained model analysis.

Module 5:Identification and Classification of diseased plants with suggested remedies.

By this phase, the trained model is integrated with the android application. In this phase the images to be tested are clicked and uploaded to the application after granting the required permissions to it. The test images are then processed according to the trained model and are classified into a particular category of identified disease which is then returned to the end user through the Application UI. The application is also fed with the information regarding the diseases classified and their remedial measures. Based on the disease classified the application fetches the relevant information   from the data fed and returns the suggested remedies to the end user via the application UI.

CHAPTER 6

TESTING

In this chapter, an overview of testing is provided to verify the correctness and the functionality of the system. Software testing is the process of analysing a software item to detect the differences between the existing and the required conditions and to evaluate the features of the software item. Software testing is an activity that should be done throughout the development process. Software testing is a task intended to detect defects in software by contrasting a computer program's expected results with its actual results for a given set of inputs.

TEST ENVIRONMENT:

THE SOFTWARE WAS TESTED ON THE FOLLOWING ENVIRONMENTS:

1. Python 3.7
2. Windows environment.
3. Android Application

Test Case:

Set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

- Features to be tested
- Purpose of testing
- Pass/Fail Criteria

6.1 UNIT TESTING

Unit testing is the testing of individual hardware or software units or groups of related units. Using the unit test plans prepared in the design phase, important control paths are tested to uncover errors within the boundary of the modules. The interfaces of each of the modules are tested to ensure proper flow of the information into and out of the modules under consideration.

Each unit in this project was thoroughly tested to check if it might fail in any possible situation. This testing was carried out at the completion of each unit. At the end of the unit testing phase, each unit was found to be working satisfactorily in regard to the expected output from the module.

TEST CASE ID	Training dataset	Test dataset	Accuracy
1	300	20	42.6%
2	400	20	51.37%
3	600	20	66.74%
4	850	20	87.00%
5	1000	20	99.15%
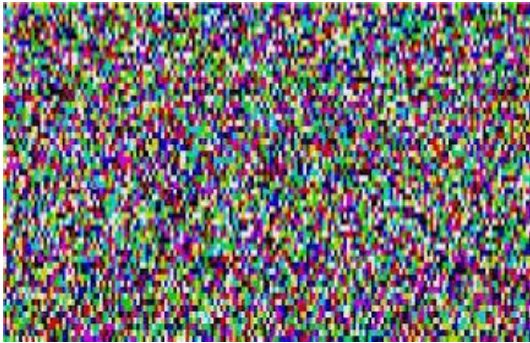
Table 6.1: Test Cases for Untit Testing

6.2 INTEGRATION TEST

Integration testing is the testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them. The various modules are tested for their accuracy and compatibility. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together.

6.3 SYSTEM TESTING

System testing is the testing conducted on a complete, integrated system to evaluate the system compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passes integration testing.

TEST CASE ID	INPUT IMAGE	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1		No disease identified	No disease identified	PASS
2		No disease identified	Classified as red rot disease	FAIL
3		Not a Sugarcane plant. Retake the image	Yellow leaf disease	FAIL
4		Classified as Eye-spot disease	Classified as Eye-spot disease	PASS
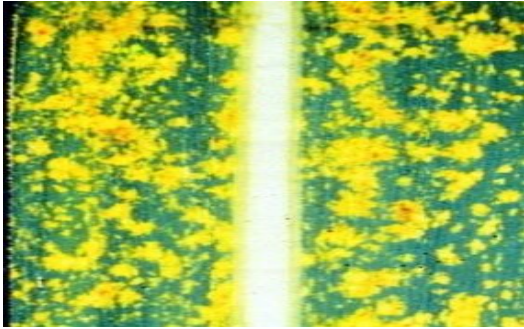
5		Classified as Red rot disease	Classified as Red rot disease	PASS
6		Classified as Yellow leaf disease	Classified as Yellow leaf disease	PASS
7		Classified as White rust disease	Classified as White rust disease	PASS
8		No disease identified	No disease identified	PASS

Table 6.2: System Test Case

SNAPSHOTS

1. LOADING SCREEN
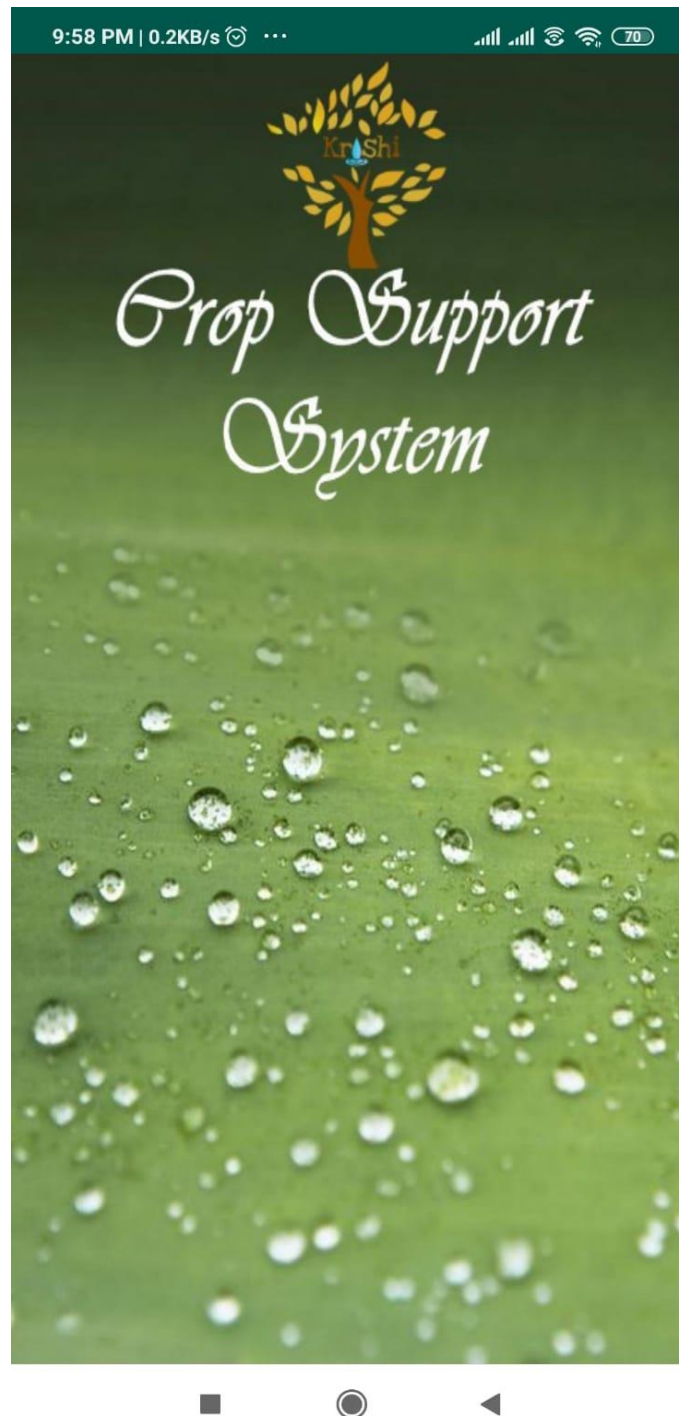
Fig 6.1: The above image shows loading screen when Android application is launched

2. HOME PAGE

Fig 6.2: The above image shows home page

3. SIDE BAR VIEW
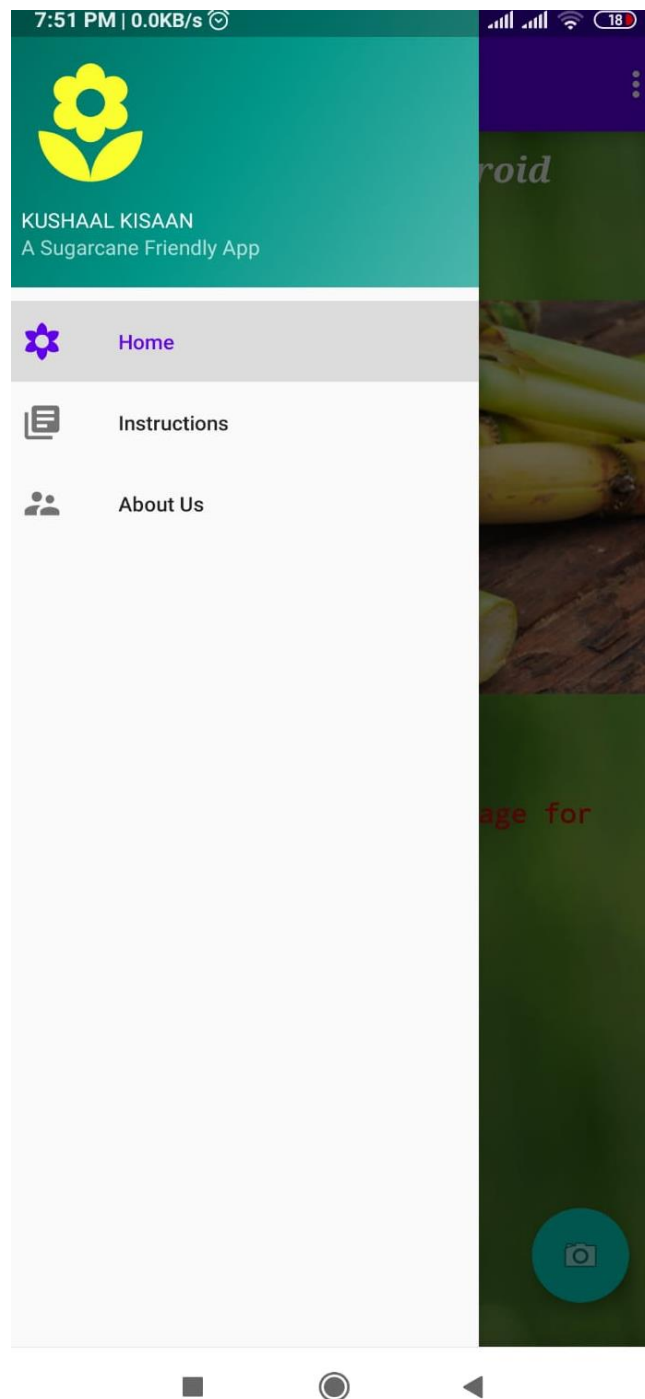
Fig 6.3: The above image shows side navigator bar with various options to select
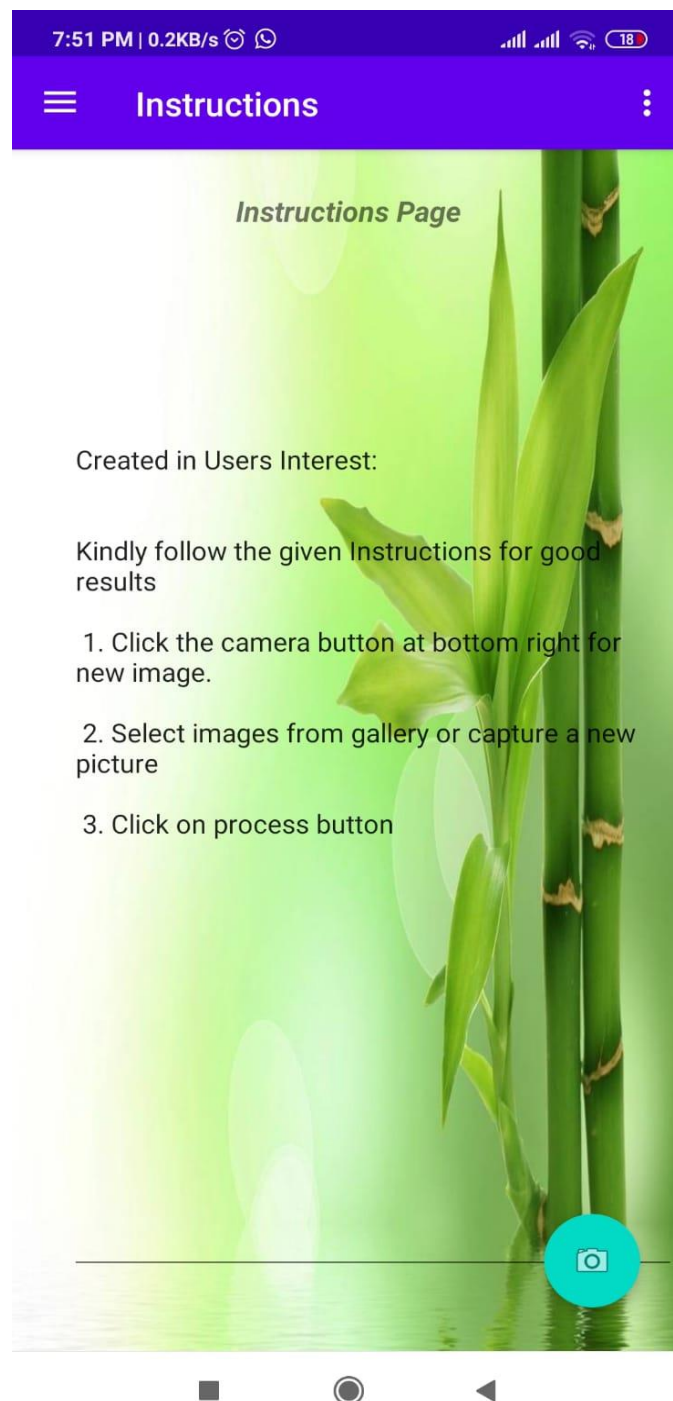
4. INSTRUCTIONS PAGE

Fig 6.4: The above image shows instructions page
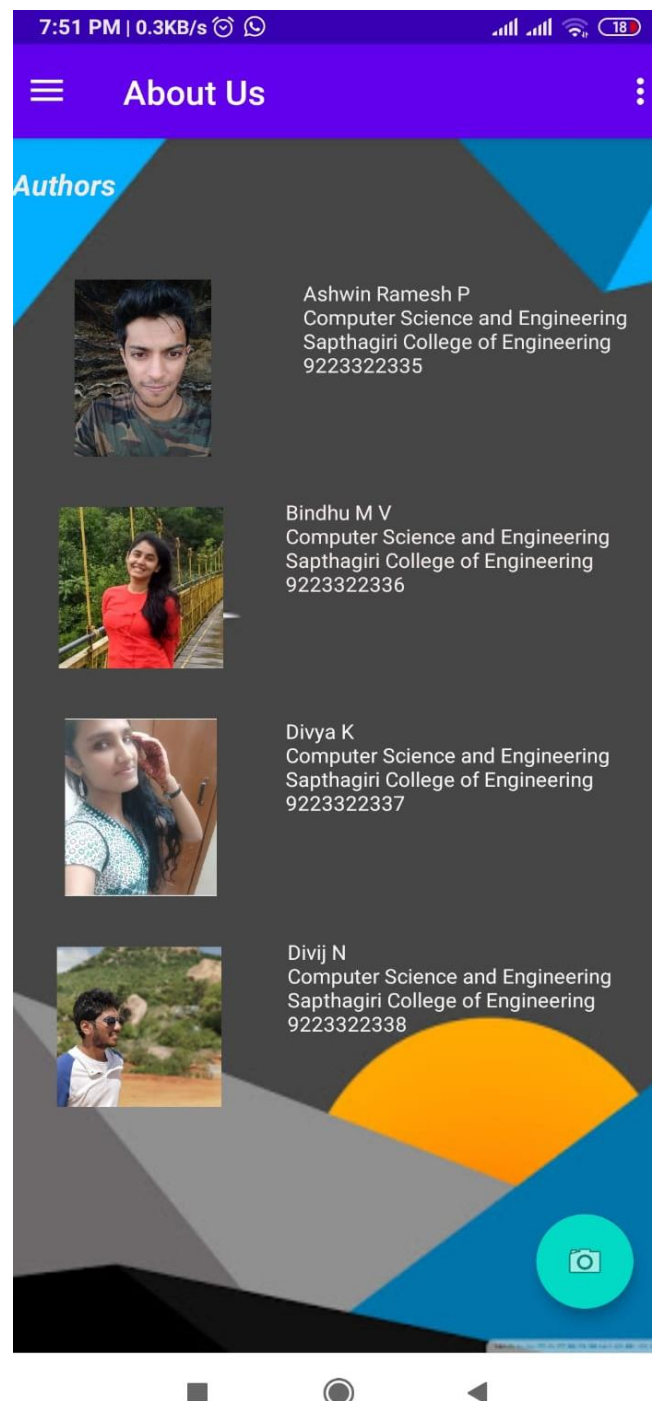
5. ABOUT US PAGE

Fig 6.5: The above image shows information of app creators
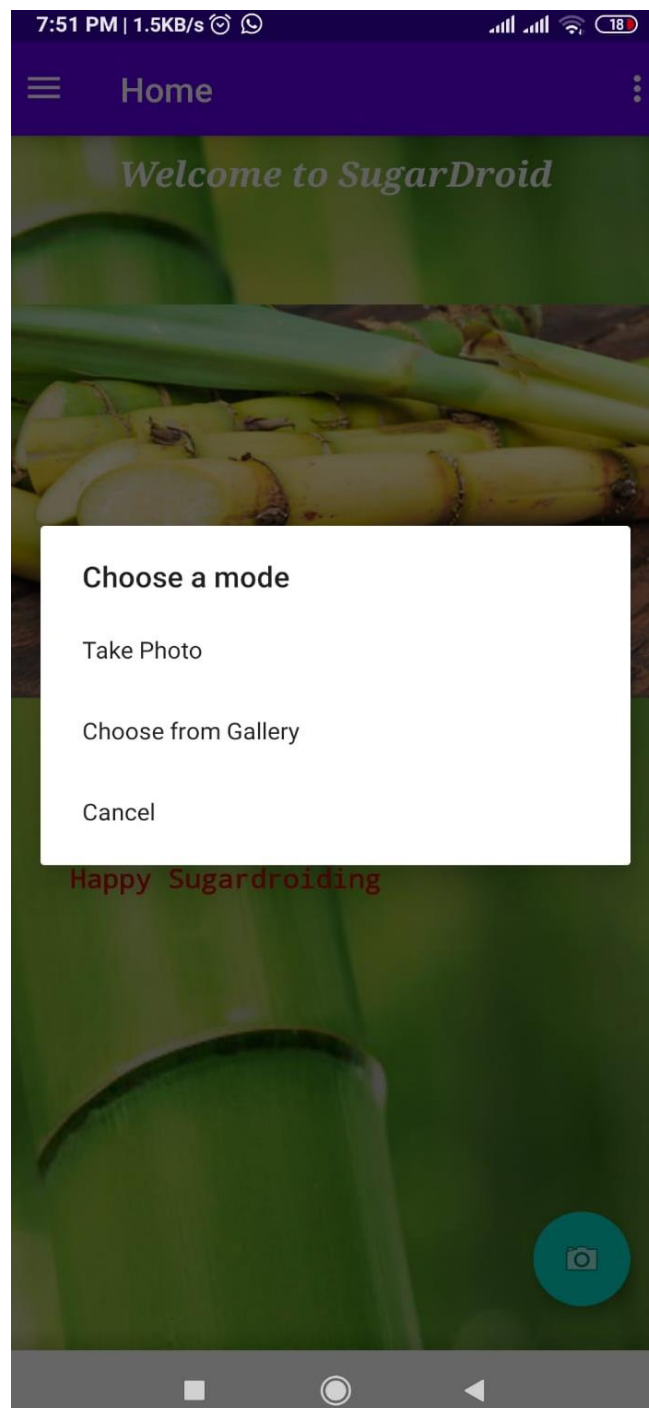
6. IMAGE SELECTING OPTIONS

Fig 6.6: The above image shows various options that can be used to upload image
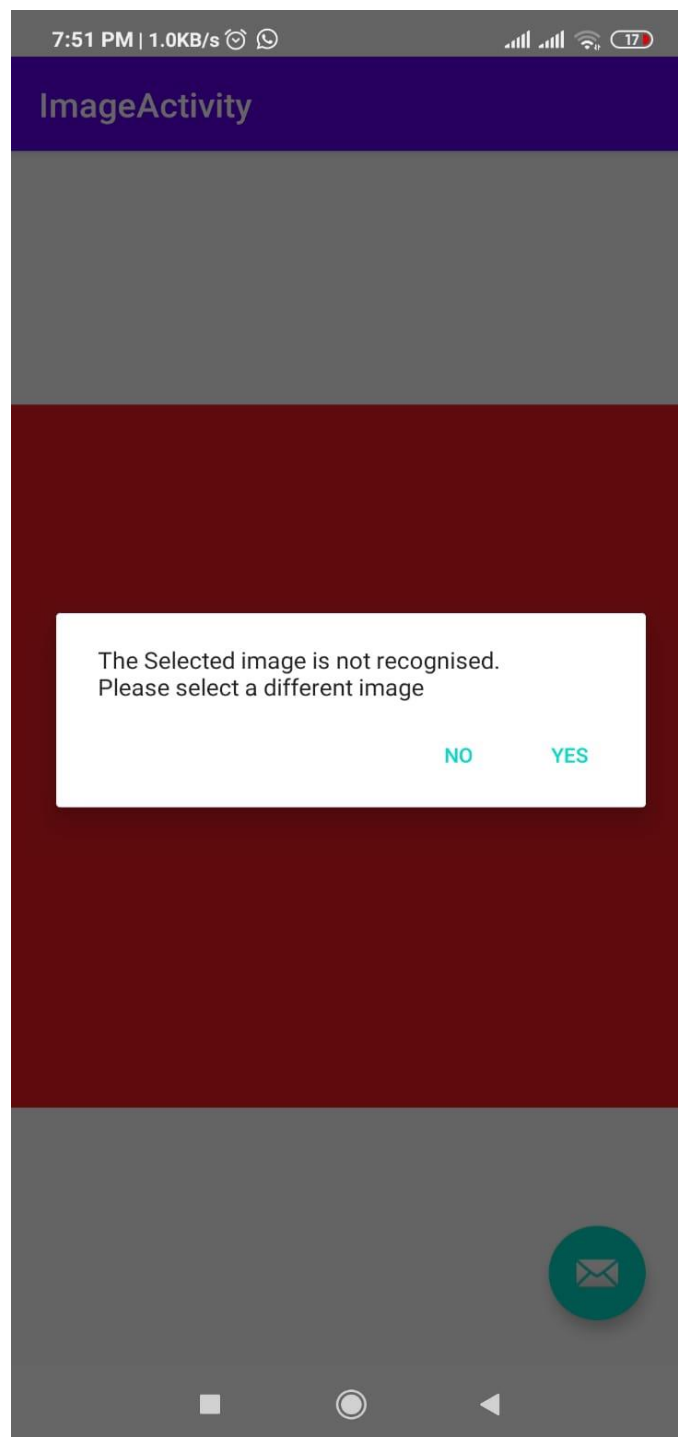
7.   WHEN WRONG IMAGE IS GIVEN FOR TEST

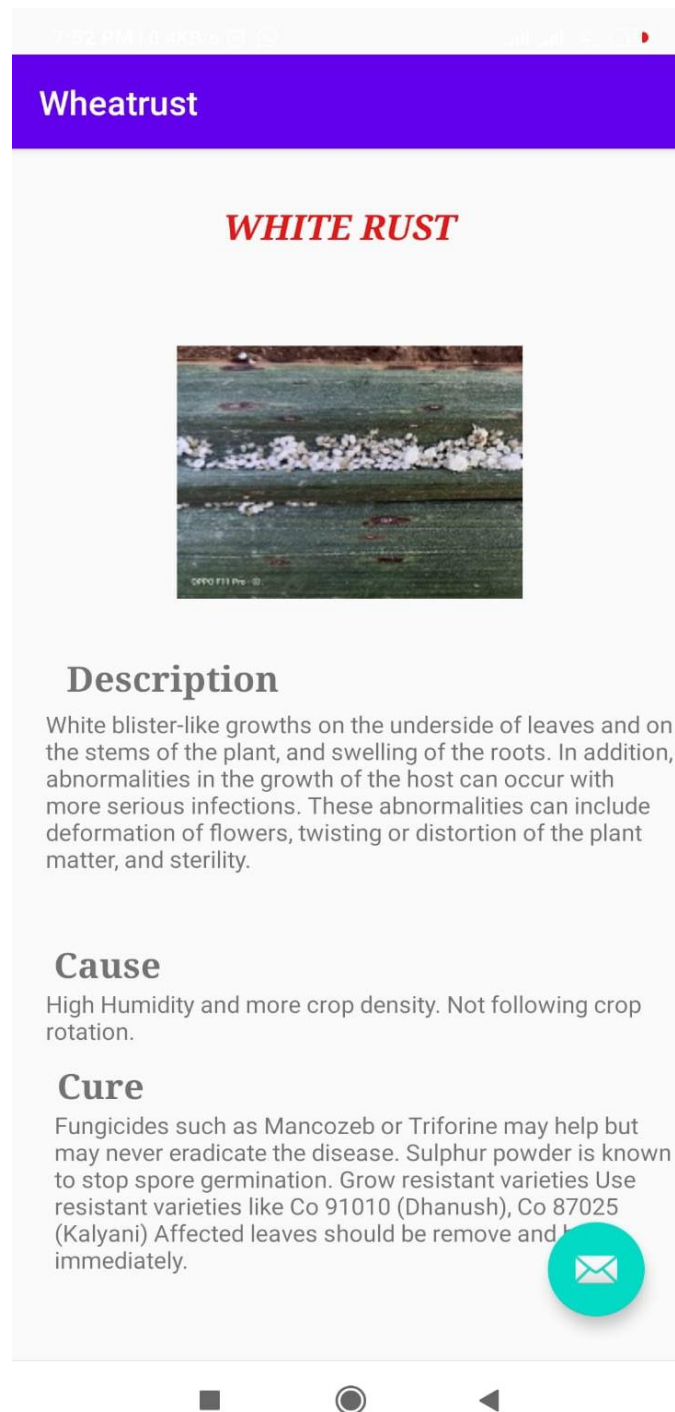Fig 6.7: The above image shows result page when wrong image is passed for test

8. OUTPUT PAGE

Fig 6.8: The above image shows output page

9. INPUT IMAGE DISPLAYED

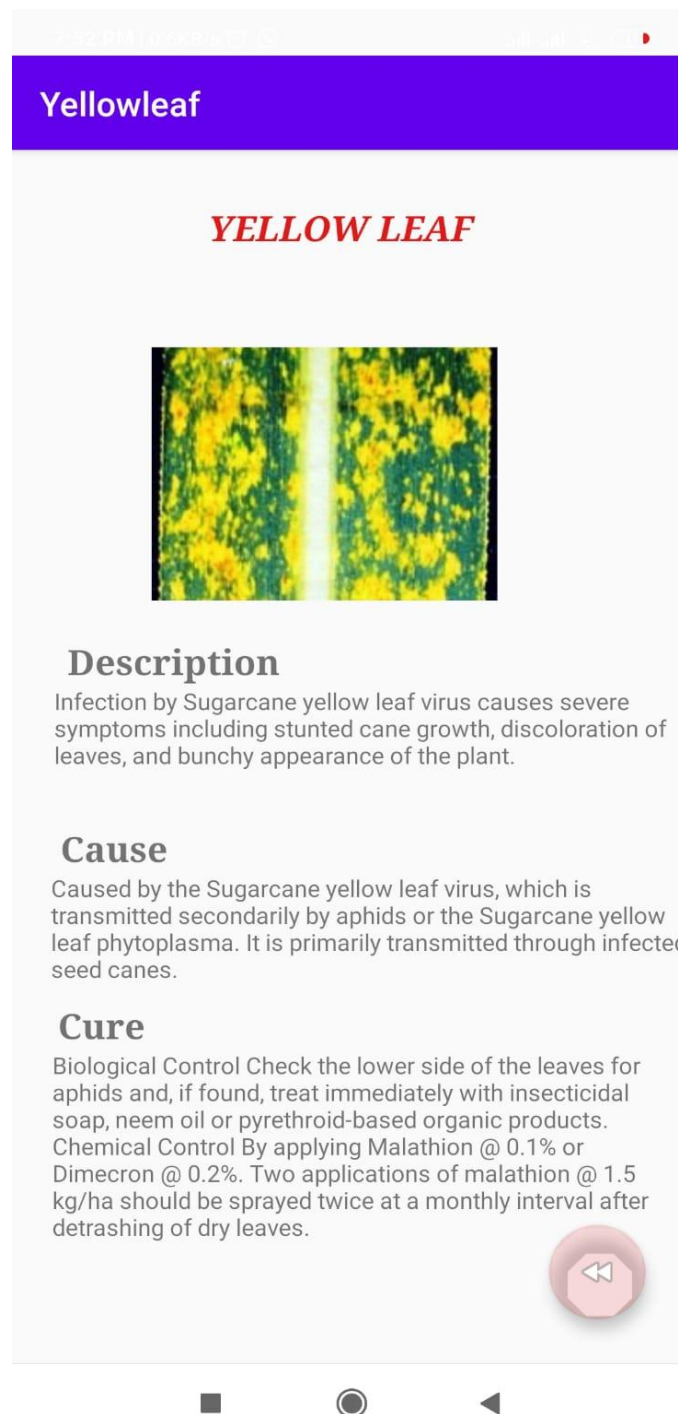Fig 6.9: The above image shows the selected image

10. OUTPUT PAGE 2

Fig 6.10: The above image shows output page

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 CONCLUSION

The problem of early stage disease identification in crops has always been a challenging task to the farmers. The proposed system is an efficient solution to address this issue. It automates the disease detection process by comparing the suspected plant features with the features exhibited by thousands of diseased plants and provides the results to farmers with a high accuracy, thus helping the Farmers to take the preventing measures at the early stage to avoid crop loss. The System is designed to be a user-friendly application with an interactive and easy to use UI. The system provides easily accessible solutions as it is a self-inference system and does not require any internet accessibility to get the solutions. Thus, it can be concluded that the system is an efficient aid to farmers (regardless of the level of experience), enabling fast and efficient recognition of plant diseases and facilitating the decision-making process.

7.2 FUTURE ENHANCEMENT

This system is used to detect sugarcane diseases using neural networks and perform on device inference. It can be adopted and enhanced by using a larger dataset for disease detection in a wider variety of plants. Also system can be enhanced with features to track the plant growth continuously. And provide customized suggestions to increase crop yield

BIBLIOGRAPHY

REVIEW THROUGH JOURNAL PAPERS

1. [1] "Novel Machine Learning Based Approach For Detection And Classification Of Sugarcane Plant Disease By Using DWT". Suyash S Patil, Sandeep Thorat. CCIP, 2016.

2. [2] "Detecting Sugarcane Borer Diseases Using Support Vector Machine". B Sravya Reddy, R Deepa, Shalini, P BhagyaDivya. IJRET Vol.4, 2017.

3. [3] "Sugarcane Disease Detection Using Data Mining Techniques". Tien Huang, Rui Yang, Wenshan Huang. Chinese University of Hong-Kong, Elsevier, 2017.

4. [4] "Paddy Disease Detection System using Image Processing". S Sathiamoorthy, R Ponnuswamy, M Natarajan. IJRAT, 2018.

5. [5] "Leaf Disease Detection Using Image Processing". RadhiahBintiZainon, 2012.

6. [6] "Plant Disease Detection Using Different Algorithms". Sujatha R, Y Shravan Kumar and Garine Uma Akhil. JCHPS Vol.10, 2017.

7. [7] "An Algorithm for Plant Diseases Detection Based on Color Features". TrimiNehaTete, SushmaKamlu. RICE Vol.10, 2017.

8. [8] "Real-time Hevea Leaves Diseases Identification using Sobel Edge Algorithm on FPGA: A Preliminary Study". MoshabElsghair, RakaJovanovic, Milan Tuba. IJAS Vol.2, 2017.

9. [9] "An Investigation Into Machine Learning Regression Techniques for the Leaf Rust Disease Detection Using Hyperspectral Measurement". NorfarahinMohdYusoff, Ilishairah Abdullah. ICGRC, 2018.

REVIEW THROUGH WEB REFERENCE

1. https://stackoverflow.com/questions/11453073/set-image-for-imageview-on-another-layout

2. https://medium.com/@hasangi/capture-image-or-choose-from-gallery-photos-implementation-for-android-a5ca59bc6883

3. https://www.quora.com/How-do-I-pass-multiple-values-from-one-activity-to-another-activity-in-Android-app-development

4. https://www.roseindia.net/android/what-is-android-application-framework.shtml

5. https://www.hindawi.com/journals/cin/2016/3289801/