National University of Computer and Emerging Sciences

Disease Detection, Mitigation and Growth Analysis of Cotton Crop Using Convolutional Neural Networks and Satellite Imagery

FYP Team

Abdul Rauf Bhatti…………19L-0948
Syed Yousaf Qadri…………19L-0909
Muhammad Qasim Bhatti………19L-0947

Supervised by

Ms. Abeeda Akram

FAST School of Computing

National University of Computer and Emerging Sciences

Lahore, Pakistan

December 2022

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: <u>Disease Detection, Mitigation and Growth Analysis of Cotton Crop Using Convolutional Neural Networks and Satellite Imagery</u>

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 12/10/2022

Student 1

Name: Abdul Rauf Bhatti

Signature: ABDUL RAUF BHATTI

Student 2

Name: Syed Yousaf Qadri

Signature: SYED YOUSAF QADRI

Student 3

Name: Muhammad Qasim Bhatti

Signature: M. QASIM BHATTI

Authors' Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract
Pakistan is considered as an agriculture-based country where agriculture contributes around 19 percent of GDP and employs around 42 percent of our labor force. With limited natural resources and the transition of the world towards applying eco-friendly practices, we aim to apply technology in agriculture in such a way that helps the farmers in properly managing their crop growth and achieving maximum yield without an excessive use of resources. In this document we have worked to identify the cotton crop diseases, their causes and mitigation techniques so they can be dealt effectively. Moreover, we have identified features that help in detecting the disease and then we have linked it to technology by identifying different algorithms that will help us in training our system. Lastly, the importance of soil moisture and nutrients and other factors in cotton growth is also highlighted.

Executive Summary

Cotton is a very essential crop for a country like Pakistan. This makes it critical to identify its diseases and their detection with technology so that it is diagnosed timely. Moreover, it is also very essential to monitor the crop growth at different stages and its dependency on the soil moisture and soil nutrients can certainly not be overlooked. As cotton is one of the main agricultural products upon which our economy significantly depends, it is important to make use of technology in helping to identify the challenges faced by this crop so that right decisions can be made at the right time by the cotton crop growers that result in an increased yield and more benefit for them and collectively for the economy.

Firstly, it is important to understand the diverse variety of diseases that this crop can suffer from. Our findings through the research papers have shown that these are significantly high in number with each disease having unique symptoms that can be detected on the different parts of the cotton plant. Secondly, it is also important to manage the diseases once they are identified. This must be done through different mitigation techniques that are also discussed in this document. Moreover, diseases can also be classified based on their cause such as Viral, Fungal, Bacterial and disease due to insects. This classification makes it easier to highlight the root cause of the occurrence of a specific disease. In addition to this, it has been observed that cotton diseases can be categorized based on the different parts of the plant where their symptoms are shown such as leaves, root and seedlings, ball rot and other parts. This study has highlighted that most of the diseases can be identified through leaves as compared to the plant parts. Furthermore, the dominance of the leaves in detection of diseases is further discussed in terms of the diverse categories it covers and how technology is integrated to make this detection efficient and feasible.

Computer Vision can be used to detect different kinds of diseases that affect the cotton plant. Analyzing the cotton plant can prove beneficial in determining the disease as certain areas of the cotton plant can change their color based upon the disease that has affected them. Once one can determine the disease, a mitigation process can be started which would be to suggest the appropriate chemical or medicine that is needed.

Analyzing soil moisture can be an important indicator of crop growth for cotton. There are multiple ways how soil moisture can be analyzed, such as, using wireless sensor networks or using the Sentinel-2 Satellite imagery estimates of current soil moisture can be made. Which can be used to suggest how much water is needed to maintain an optimum level of soil moisture. As, keeping the soil moisture constant for the cotton crop can result in a better yield.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Purpose of this Document

Cotton shares a major part in the total exports of Pakistan; in the Fiscal Year 2022 textile exports accounted for around $19.33B. Furthermore, as Pakistan attempts to transition towards an industrial based economy, the textile sector which is already one of the biggest export markets for Pakistan thrives upon sustained and regular production of Raw Cotton.

A lot of Raw Cotton crops in Pakistan are affected by the Cotton Leaf Curl Virus which has historically caused and continues to cause immense losses to the cultivators. Every year, Pakistan faces approximately 30% cotton crop loss due to the Cotton Leaf Curl Virus alone. This makes it critical to identify the cotton diseases and their detection using technology. Early detection of the virus can effectively help to mitigate the number of damages to the plants. Using Object Detection, we intend to identify the diseases by analyzing the cotton leaf; the leaf of the cotton plant can be analyzed to check for any possible disease that may be affecting the plant, thus, by exaggeration the whole farm. Whilst, with the growing use of technology in the farming industry, there is still less emphasis on how to detect the disease early and then help mitigate those damages. In addition, in Pakistan, this is even lower. As there is not much emphasis on employing technology-oriented solutions. Moreover, it is also very essential to monitor the crop growth at different stages and its dependency on the soil moisture and soil nutrients. Our project aims to fill this gap by building a smartphone application which will help the farmer in identifying and mitigating the diseased crop. Furthermore, another important aspect of this project involves crop monitoring to aid the produce growth cycle.

1.2 Intended Audience

The audience that we aim to target in this case are mainly the people that are involved in the cotton crop growth process. These are mainly the cotton crop farmers and other people that examine the cotton crop growth such as researchers. Cotton crop farmers can be categorized into three main categories. These are small-scale, medium scale and the large-scale farmers.

A large-scale farmer usually has abundant resources combined with high capital to identify the crop diseases and make sure that their growth is on time. This type of farmer has a high flexibility of utilizing his resources to ensure that the crop being grown is free of any disease and has a proper growth and even if it suffers from any problem then he/she can bear the expenses of hiring experts to diagnose the disease and eventually deal with it. On the other hand, medium to small scale farmers fall into the category where they have a moderate or very limited number of resources for the crop growth. Approximately 90% of the farmers in Pakistan fall into the category of small scale. They cannot bear any major disease or growth problems related to their crops as they do not have enough resources to prevent or solve the problem. Any major problem related to their crop might result in a total loss in terms of both individually for them and collectively for the economy. We aim to fill this gap by developing an application that can help in them an access to our application which will aid in enabling the farmer to detect, diagnose and deploy mitigation techniques on the diseased plant. Furthermore, the application will also enable the farmer to monitor their plantation.

1.3 Definitions, Acronyms, and Abbreviations

SDG: Sustainable Development Goal

Chapter 2: Project Vision

2.1 Problem Domain Overview

Over the years, much emphasis has been placed on developing sustainable models. Models that are efficient and that curb or curtail resource wastage. Ideas to sustainably develop and enhance many domains have been aired and talked about in recent times. With the dawn of the 21st century, work has been put in researching methods to modernize the prevalent agricultural techniques. Researchers have tried to find new sustainable techniques to grow produce and curb resource wastage. In Pakistan, the agricultural techniques are to a large extent, traditional. The current agricultural scene in Pakistan relies heavily on word of mouth and experience of others— creating room for a high degree of uncertainty by introducing delays in detection and mitigation efforts. Over the years, the term 'precision agriculture' has gained much importance. Precision agriculture, in nature, endeavors to curtail resource wastage, whilst maximizing produce. However, another important aspect of precision agriculture is process automation. In our project, we achieve this by automating disease detection and suggesting the relative mitigation techniques.

2.2 Problem Statement

Albeit an important cash crop, the cotton crop is prone to diseases. 90% of the diseases inflicted upon the cotton crop are recognizable from the crop leaf, however, the human eye can only tell so much. There is a significant chance that many of the diseases go unnoticed or are confused for others. Furthermore, cotton has been at times labeled as one of the most water demanding crops, therefore, it is imperative to explore methods to efficiently manage the inundation cycle of the cotton crop plantation. This project will endeavor to address this problem by deploying modern computer imagery techniques to identify the diseased plant, the relevant disease and suggest mitigation techniques.

2.3 Problem Elaboration

As mentioned in the preceding section, one of the biggest challenges of cotton crop farming includes tackling the diseases that harm the crop. The cotton crop can suffer highly due to these diseases and the output can shrink greatly in size. Furthermore, 90% of these diseases can be attributed to the colour, texture and general outlook of the crop leaf, making room for ample digitization. Many of the cotton diseases appear on the leaf in the form of 'spots', making it difficult at times for the human eye to tell them apart. If correction is applied to the diseased crop, there are high chances of recovery. Therefore, in this project, we have reviewed several techniques that have been used in detecting diseases in cotton and/or other crops. Before proceeding to the various techniques that help in disease detection in cotton crops, it is important to recognize and state the possibilities from which the diseases can be detected. The possible areas which can help in diagnosing a diseased cotton crop include:

1. Leaf
2. Stem
3. Cotton Plant
4. Seed Composition (Genetic disease)

2.3.1 Dataset Collection

From the above-mentioned classes of cotton diseases, the dataset for leaf-based diseases is widely available. As part of our project and initial research, we were unable to find a similar dataset for stem based or seed-based diseases. It was also discovered that genetic diseases that take place in the seed composition are best recognized inside a lab environment with chemical aids, the same also holds true for stem-based diseases.

Focusing on leaf-based diseases, we were able to come across research studies that quoted testing a wide number of relevant pictures on their proposed model and methodology, indicating clearly that datasets for leaf-based diseases are available on the internet. Furthermore, as part of our project, we identified multiple sources to source leaf data (images) from. These sources include Kaggle, GitHub, Plant Village etc. However, another relevant challenge pertaining to the dataset would be of coverage. It would be imperative to know how many diseases (of the ones recognizable from the leaf) are being covered by the datasets that are available online. Once the dataset has been gathered, using an appropriate methodology for prediction and training is another challenge.

2.3.2 Dataset Preprocessing

The gathered dataset will not need to be preprocessed as part of this project. We will endeavor to gather already categorized data from various sources that contain the relevant images. However, as part of the imagery process, the dataset might need to be preprocessed to account for environmental changes of the captured image.

2.3.3 Segmentation

In both cases, image processing steps involving image segmentation would be needed. Inside the segmentation process, it would be important to have a methodology that accurately differentiates the background from the target area. Furthermore, it would have to segment the target area further to accurately recognize disease ridden areas before classifying them on the trained model. Segmentation, or feature extraction will allow the program to focus on the target parameters before passing them on to the classification model.

2.3.4 Classification

Any solution based on computer imagery will be as good as the method used to classify the input images on the already trained model. Therefore, an essential part of this project was to analyze and correctly recognize the best classification methodology for implementing such. These methodologies can be grouped into two broad categories:

1. Traditional Imagery Techniques
2. Deep Learning Techniques

With traditional techniques and deep learning techniques, the steps prior to classifying the data are similar. Deep Learning Techniques comprise of Convolutional Neural Networks, whilst Traditional Imagery Techniques are composed of a number of techniques i.e. Support Vector Machines, Artificial Neural Networks and Multi Support Vector Machines (to name a few). In the quest of searching for the best possible methodology, techniques used in other crops for a similar or matching purpose could also prove helpful.

2.3.5 Mitigation

Once the disease has been detected and correctly recognized, another important part would involve suggesting mitigation techniques to the farmer. Techniques to mitigate the damage inflicted on the cotton crop are an essential area of this project. It is important to suggest to the farmer what to do once a disease has been recognized. Consequently, techniques need to be researched and documented that pertain to a certain stratum of diseases. Once any of the disease has been recognized, the project should suggest the correct mitigation technique to the farmer.

2.3.6 Continuous Monitoring

Besides disease detection and mitigation, an important aspect of the cotton crop growth cycle involves continuous monitoring of the cotton crop. This is done to ensure that the cotton crop is growing at an acceptable and satisfactory rate. Moreover, monitoring of this kind can also help in supplementing disease detection and mitigation. However, crop monitoring is a rather more complex task than disease detection. To monitor a crop effectively, several parameters need to be recognized, that are to be monitored at fixed intervals or continuously. These parameters can include:

1. Soil moisture levels
2. Nutrients level in plantation soil
3. Crop height monitoring
4. Crop product monitoring (boll output)
5. Boll analysis
6. Leaf growth analysis (when first, second or other leaves are grown by the cotton crop)

Further parameters can be explored in any of the major parameters written above. However, the methodology to be used remains as the major question that needs answering. There are several possible methodologies. Firstly, a sensor network of planted field sensors could be used to gather and analyse the relevant data. Furthermore, this data could be transferred to the farmer using the application as a help to better their crop yield. Another possible idea for effective monitoring could include the use of drones or robots. Drones have been widely in use in precision agriculture (a type of agriculture technique that involves the use of advanced methods to regulate and help in effectively growing produce). They have helped in monitoring crop health from above the field and are preferred as they help in covering a relatively large area in a very short amount of time. However, the extent of information that can be grasped from such height and distance is a point of contention. One other and an increasingly popular monitoring technique involves the use of satellite imagery. Satellites such as The Sentinel Copernicus 2 provide data such as the soil moisture index, NDVI (Normalized Difference Vegetation Index) that can help in approximating soil composition and the resultant yield of the planted crop. Furthermore, since the cotton crop is one of the most water demanding crops, it is imperative to maintain a sufficient water level. The water demand of the crop plant creates another optimization problem for a country like Pakistan.

2.3.7 Prediction

Pakistan has been labelled as one of the most water stressed countries on the planet. Therefore, the need to efficiently manage water resources is amplified. The cotton crop has been labelled by some as having a ‘linear’ relation with the inflow of water, which clearly cements the reliance of the cotton cash crop on water. By using the data extracted from the satellite imagery alongside weather trends of the plantation area, sufficient prediction and analysis could be done on the level of water that is required to sustain the cotton's reliance on water. Furthermore, a linear regression model can be trained, based on the satellite data and trending weather patterns to sustain an appropriate crop water cycle.

2.4 Goals and Objectives

- Study the correlation between the many diseases and the cotton crop leaf.
- Study prior computer imagery methods and techniques that have been researched regarding disease detection in cotton and/or other crops.
- Study and review the mitigation techniques linked to the diseases and suggest the possible course of action.
- Study the correlation between crop yield and field soil moisture.
- Study and obtain different cotton crop data sets available on Kaggle and similar research papers.
- Research the use of satellite imagery to gather data such as the field soil moisture.
- Study the correlation between soil moisture and weather patterns.
- Produce an android application that allows the user to gather infected leaf photos and diagnose the disease with mitigation suggestions.
- Help the user to analyse the amount of water required to maintain a satisfactory level of soil moisture in cotton plantation.

2.5 Project Scope

The project will consist of reviewing computer imagery techniques that have been researched regarding plant disease detection and mitigation. Furthermore, a key area of this project will rely on implementing the chosen strategy in the form of an android application. The android application in question will stand to be an implementation of the reviewed research work. Therefore, it will endeavour to also address the management of soil moisture and enable farmers to manage the water resource efficiently, thereby curbing water wastage.

2.6 Sustainable Development Goal (SDG)

This project targets two of the SDG goals. It targets Industry Innovation and Infrastructure by digitizing the diagnosis process of a diseased cotton crop. A large part of the agricultural scene in Pakistan relies on the expertise of others, which can at times be ill-founded. Therefore, this project aims to curb such problems and diagnose with maximum possible efficiency the probable disease that inflicts a cotton plant. Furthermore, the app will suggest possible courses of action and mitigation techniques so that the impact of the identified disease can be lessened, and the output can be maintained.

The other SDG goal this project addresses are responsible consumption and production. By monitoring the soil moisture through utilizing data generated by satellite imagery, the project will help in curbing water wastage. It will analyze short-term weather predictions and by establishing sufficient correlation between soil moisture and predictions, it will suggest times when the crop could use additional inundation, thereby responsibly and efficiently managing water resources in an already water-stressed nation.

Chapter 3: Literature Review / Related Work

3.1 Definitions, Acronyms, and Abbreviations

Abbreviations and Acronyms

CNN: Convolutional Neural Network

ANN: Artificial Neural Network

SVM: Support Vector Machines

GLCM: Gray-Level Co-Occurrence Matrix

RFFD: Rust Foliar Fungal Disease

ALSFD: Alternaria Leaf Spot Fungal Disease

BPNN: Back Propagation Neural Network

GMCD: Grey-Mildew Cotton Disease

NDVI: Normalized Difference Vegetation Index

PCA: Principal Component Analysis

Definitions

Classification: ML technique of classifying an entity according to the nearest similar type of entity.

Segmentation: Computer Imagery technique used to divide up a target image. Usually done to separate different features e.g. background from foreground.

Extraction: Targeted image segmentation that involves taking specified information from the cotton leaf.

3.2 Detailed Literature Review on Cotton Crop Disease Identification and Mitigation

In this section most of the cotton diseases have been identified along with their symptoms to facilitate their detection. Moreover, management techniques have also been identified once a specific disease is detected to reduce its effect and decrease the loss. Furthermore, cotton diseases are categorized based on whether the disease is Fungal, Bacterial, Viral or Insect based, and another categorization is done based on the parts of the plant that are affected by the disease. Lastly, importance of cotton disease detection through the leaves is highlighted with multiple justifications.

3.2.1 Ayub Agricultural University Report

Cotton plants consist of a main stem that gives rise to multiple branches [1]. The leaves of the plant spirally surround the branches and have 3-5 triangular lobes petioles. Pakistan has a major share in the cotton growth sector. Therefore, research on cotton crops and its diseases are very essential. According to the research of [2] Ayub Agricultural Institute, Faisalabad, there are various types of cotton diseases that affect the cotton crop's growth and yield in Pakistan. One of them is Twig and Stem Blight of Cotton [2], [3] where the decay starts from the twigs of the plant and then propagates to the leaf of the plant eventually causing its death. Secondly, cotton suffers from Ascochyta blight whereby its leaves show gray or brown spots that are encircled by a red halo. Another symptom of this disease is that the plant's stem is covered by reddish purple cankers. [2], [3], [4]. Another cotton crop disease is Fusarium wilt [2], [3] whereas the name suggests, wilting of cotyledons and seedling leaves occurs. Here the vascular system of the plants is attacked, and it fades in its color. Moreover, cotyledons turn chlorotic at the edges and then necrotic. Bacterial Blight of Cotton [2], [3] is another disease where the entire part of the plant above the ground is attacked. Angular spots are shown on the leaves and the ball rot [1]. Water-soaked spots appear on the leaves' lower surface initially followed by the other surface later. These spots turn yellow and expand in size [1], [2]. One of the most famous cotton diseases known is the Cotton Leaf Curl Virus. Upward and downward curling of leaf is shown [1]. This disease can be easily spotted through its leaves as they curl in their size followed by darkened veins when viewed from below against the sunlight and vein swelling [1], [2]. Eventually, there is a cup shaped leaf enation

3.2.1.1 Critical analysis of the research item

There are several diseases that can not only be detected through the leaves, rather they are to be detected from other parts of the plant such as stem, root and ball rot. In this case the leaves fail to detect those types of disease. However, the diseases detected through leaves have a major share in cotton diseases for instance, leaf curl virus is a very major disease suffered by the cotton crops in Pakistan and as the name suggests, it can easily be detected through the leaves of the plant. Majority of diseases suffered by the cotton crops in Pakistan can be mostly detected through the leaves.

3.2.1.2 Relationship to the proposed research work

This paper helps us to highlight the major diseases faced by the cotton crop in Pakistan. As our application will also be based mostly for this geographic region, it indicates to us on what diseases to focus the most on.

3.2.2 Plant Village Report

This paper shows that the cotton diseases can be classified into 4 sub-categories that are mainly Fungal, Viral, Bacterial and Insect based diseases. This article also further aids in guiding the management techniques so that the disease effect can be prevented or minimized. The classification categories and the relevant diseases along with their mitigation techniques are shown in the following table.

Table 1: Fungal Disease Identification and Mitigation Techniques
The following presents the different fungal diseases and suggests possible mitigation techniques that follow.

Disease Name	Symptoms	Mitigation Technique
Alternaria leaf spot	This disease portrays reddish-brown spots on the leaves as well as ***cotyledons*** that grow in size and create a centralized pattern. Necrotic areas might have purple textures and the centers of lesions usually drop out after	Plowing of the crop residue into the soil to decrease inoculum intensity; provision of adequate irrigation and nutrients, specifically potassium.

	drying and create a hole in the leaf causing a "Shot " hole appearance of it.	
Asochyta blight	(explained in the previous section)	Plowing crop debris into soil after harvest; crop rotation has negligible effect on control of disease.
Cercospora leaf spot	Circular red lesions on leaves that expand and change to gray or white at the center.	Plowing of the crop residue into the soil to decrease inoculum intensity Provision of adequate irrigation and nutrients Applications of relevant foliar fungicides can be needed on susceptible cultivars.
Fusarium wilt	(explained in the previous section)	Use the authentic, disease-free seed. Fumigating the soil may reduce disease incidence.
Target spot	Symptoms of this disease can be shown on leaves, ***boll bracts*** and ***bolls*** as dark chocolate dots and the leaves retain their green color.	Use available resistant varieties. Follow crop rotation. Spray suitable fungicide.
Root Rot	The disease specifically targets the roots of the plant. Wilting causes fall of leaves and shredding of roots.	

Table 2: Bacterial Disease Identification and Mitigation Techniques
The following table presents the different bacterial diseases and suggests possible mitigation techniques that follow.

Disease Name	Symptoms	Mitigation Techniques
Bacterial Blight of Cotton	(explained in the previous section)	The use of resistant cotton varieties is the most effective method of controlling the disease; cultural practices such as plowing crop residue into soil after harvest can also limit disease emergence.

Table 3: Insect Related Disease Identification and Mitigation Techniques

The following table presents the different insect diseases and suggests possible mitigation techniques that follow.

Disease Name	Symptoms	Mitigation Techniques
Aphids	Tiny insects can be spotted on the lower/underside of the leaves and/or stems. In case of heavy infestation, the leaf turns yellow with necrotic spots.	insecticidal soaps or oils such as neem or canola oil are usually the best method of control. Reflective mulches such as silver colored plastic can deter aphids from feeding on plants; sturdy plants can be sprayed with a strong jet of water to remove aphids from leaves; insecticides are generally only needed to treat aphids if the infestation is very high
Amyworm	It shows big holes in leaves due to intense feeding of larvae. Holes are also present in ***bracts*** associated with ***bolls***. Dry wounds may be shown the the fruit and clusters of eggs may appear on the leaves.	Use of Bacillus thuringiensis; Biological control by natural enemies which parasitize the larvae
Cotton Bollworm	Holes are chewed in bases and the insects appear around the holes. Caterpillars may be white or yellow-green in color. Eggs are laid on the upper and the lower leaf surface and change color from cream white to brown red ring before hatching.	Bacillus thuringiensis or Entrust SC may be applied to control insects on organically grown plants

Table 4: Viral Disease Identification and Mitigation Techniques
The following table presents the different viral diseases and suggests possible mitigation techniques that follow.

Disease Name	Symptoms	Mitigation Techniques
Cotton Leaf Curl Virus	(explained in the previous section)	Vector control reduces CLCuVD infestation.Seed dressing insecticides, yellow sticky traps, Adulticides / Nymphicides are effective tools to suppress whitefly.Resistant variety is the best solution

3.2.2.1 Critical analysis of the research item

By analyzing the data of this research paper, it can be depicted that quantity wise, the fungal class has most of the chunk associated with the cotton diseases. Furthermore, insects also have a role in attacking the cotton plant which is not highlighted in any other research paper used in this document. Moreover, although quantity wise the bacterial and viral diseases are low in number, impact wise these are considered one of the major diseases of cotton.

3.2.2.2 Relationship to the proposed research work

This information will help us link the diseases to its relevant class. This way, the information is more organized and classified so it is easier for the system to narrow down the cause of the disease and more detailed information can then be provided to the end user by narrowing down the root cause of the disease through this classification.

3.2.3 Dekalb, Asgrow, Deltapine Report

This research paper also classifies the cotton diseases but this time the classification is done according to different parts of the plant. These parts are namely Seedling, Foliar, Stem and Root, Ball rots and lastly Nematodes.

Classification on the basis of different parts of the plant are shown below:

1. **Seedling Diseases**:
   a. This includes Root Rot diseases (Black Root Rot and Pythium), Fusarium and Rhizoctonia.
2. **Foliar Diseases**:
   a. **Ascochyta blight**: (Mentioned above)
   b. **Bacterial Blight**: (Mentioned above)
   c. **Alternaria Leaf Spot**: (Mentioned above)
   d. **Stemphylium Leaf Spot**: Foliage instantly turns red as lesions have purple margins with gray centers. “Shot Holes” appear on leaves where lesion centers are detached from leaves. [4]

   e. **Cercospora Leaf Spot**: (Mentioned above)
   f. **Cotton Leafroll Dwarf Virus** (CLRDV): Symptoms are usually shown at the top of the plant. Leaves turn red, curl and limp. Leaf growth becomes stunted a soon as reddened leaves are spotted. [4]

3. **Stem and Root Diseases**:
   a. **Cotton Root Rot**: (Mentioned above)
   b. **Fusarium wilt**: (Mentioned above)
   c. **Verticillium Wilt**: Leaves become randomly crippled as the disease spreads. Moreover, some parts of leaves remain green whereas other parts turn brown and die. Stem can be chopped to see brown-black vascular tissue caused by the disease. [4]
4. **Boll Rots**:
   a. Boll rots first appear soaked and brown. On severe attacks the ball rots become dark brown. [4]
5. **Nematodes**:
   a. **Root Knot**: Root galling is exposed upon digging up the plant. Affected areas of the fields are in segments and can be stunted and chlorotic. [4]
   b. **Lance**: These usually feed on the root tips and might result in more branching in the lower area of the plant close to soil. Plants also suffer from stunted chlorosis. [4]

3.2.3.1 Critical analysis of the research item

This paper divides the diseases into different categories based on the part of the plant that is attacked by each disease. It is analyzed that majority of the diseases occur on the foliar area followed by stem and root, nematodes, seedlings and ball rots. It can be depicted through this information that most of the diseases show their symptoms on leaves as foliar area secured the majority and this is followed by stem and roots.

3.2.3.2 Relationship to the proposed research work

This classification helps our research prioritize what areas of the cotton plant are most likely to be attacked by a disease which in this case are leaves of the plants and then the stem and roots followed by other parts. This will enable us to allocate our research on the cotton plant parts accordingly so that we can accurately detect the disease part wise that adds to the efficiency of our system.

3.2.4 Detection of Diseases on Cotton Leaves and its Possible Diagnosis

The main source for the disease is the leaf of the cotton plant. Around 80-90% of diseases exist on the leaves of the cotton plant [5]. Study of diseases on the cotton leaf can be studied through image processing. A naked human eye cannot catch the difference in minute variation of the image of infected cotton as that minute variation may depict the presence of a different disease on the cotton leaf. Diseases on the cotton leaves can be classified into Bacterial, fungal, viral, and diseases caused by insects [5]. All these diseases severely affect the cotton leaf once the plant suffers from the disease. We can use image segmentation to examine the diseases on the cotton plant by scanning the cotton leaves through our relevant scanning device. Moreover, the features of the leaf image indicate that whenever crops suffer from a disease, spots can be observed on the leaves and the leaf spots are acknowledged as the most important units that depict the existence of the disease and are known as the identifier of the crop's disease. Diseases can then also be classified on different leaf samples through a set of spot features used to detect and classify the different disease leaves. Infected part of leaves shows variation in color (RGB values) for different diseases and the combined RGB value differs for each disease.

3.2.4.1 Critical analysis of the research item

Idea of leaves being the central part of detecting cotton disease has been the major outline of this paper. It states that around 80-90% of the cotton diseases are identified through leaves where they are bacterial, fungal. Viral or insects based. All this shows the dominance of leaves for disease detection in comparison to other parts that according to these statistics only account 10-20% collectively.

3.2.4.2 Relationship to the proposed research work

Information stated in this paper emphasizes again on the dominance of leaves and their role in showing symptoms of the majority of the diseases. Our main target for disease detection will now be focused on leaves and their datasets to train our systems as this way maximum diseases can be diagnosed. It also indicates that there are still some diseases that are only detected through other parts so emphasis will also be laid on those parts as well to achieve accuracy and efficiency.

3.2.5 Literature Review Summary Table

The columns in the table depend upon your problem and should be specific to your project.

Table 5: Identification of Cotton Diseases and Mitigations

This table encompasses the sources that were used to write this section of the detailed literature review.

No.	Name, reference	Author	Year	Description
1.	Green Group, [1]	Green Crop Ltd.	2018	This article mainly discusses the major cotton diseases in Pakistan along with their causes and symptoms.
2.	Ayub Agricultural Report, [2]	Ayub Agricultural Research Institute	2020	Research done on different cotton diseases in Pakistan highlights their symptoms and identifies their management techniques.
3.	Plant Village, [3]	Pennsylvania State University	2018	Cotton diseases are classified into different categories Viral,Bacterial, Insects etc.
4.	Dekalb, Asgrow, Deltapine, [4]	Bayer Crop Science	2021	Cotton diseases are categorized on the basis of different parts of the plant that are affected when the disease attacks it.

5.	Detection of diseases on cotton leaves and its possible diagnosis, [5]	Viraj A Gulhane, Ajay A. Gurjar,	2011	Detection of diseases on cotton leaves and its scalability and feasibility is discussed. Emphasis is laid on the dominance of cotton leaves for disease detection.

3.2.6 Conclusion

According to the given references, it is clearly shown that majority of the cotton disease symptoms are shown on the leaves of the plant approximately 90% [5]. This statistical figure clearly shows the dominance of cotton leaves being used to determine the types of the disease. However. there are several diseases that cannot clearly be detected through the leaves, rather they are to be detected from other parts of the world plant such as stem, root, and blot. In this case the leaves fail to detect these types of disease that clearly show the weakness of this method. However, quantity wise and significance wise, these diseases do not have a major share in cotton disease problems as they are comparatively rare as compared to the diseases that can be detected through the leaves. Moreover, the diseases detected through leaves have a major share in cotton diseases for instance, leaf curl virus [2] is a very major disease suffered by the cotton crops in Pakistan and as the name suggests, it can easily be detected through the leaves of the plant. Majority of diseases suffered by the cotton crops in Pakistan can be mostly detected through the leaves. As there is an abundant number of diseases of cotton detected through leaves, there is often a similarity between diseases in terms of their symptoms shown on the leaves. It is very important to classify the diseases based on these symptoms as only a slight variation of leaf symptoms can differentiate between the diseases. Human eye is incapable of capturing this difference as the complexity is high. Although it is shown that many diseases in cotton can be captured through leaves, classifying leaf images for different diseases is a big challenge itself.

3.3 Detailed Literature Review on Disease Identification of Cotton Leaves Using Advanced Image Processing and Convolutional Neural Networks

The use of image processing techniques in agriculture has certainly been a growing phenomenon. It has, over the years, attracted enough attention from researchers all around the world. In recent times, much emphasis has been placed on detecting diseases that harm cash crops. Pakistan, being an agricultural country (much like India), is in dire need of further digitization of the agricultural sector and the usage of such techniques in practical farming could certainly lead the way in changing the outset of the agricultural scene in Pakistan. The constraints of the human eye are another leading factor for ongoing research in the above-mentioned field. The cotton crop can attribute 90% of its diseases to the coloring and texture of its leaf. Human eye, being as magnificent as it is, can certainly fall short of expectations at times. This is where room for image processing is created. Image processing methods can help in better diagnosing the prevalent disease in a crop. Various methods have been tried, researched, and put in practice over the years. The fundamentals, although, remain much the same. The main components of disease detection through computer imagery include preprocessing, segmentation, extraction, and classification. All of these have been defined in the preceding section concerning definitions and abbreviations.

3.3.1 Disease Detection of Cotton Leaves Using Advanced Image Processing

In this research item, work has been mainly done to use image processing to process and classify images of cotton crop leaves. The main goal behind this research is to match the diseased cotton leaf (and therefore the associated plant) with the correct disease. In the mentioned research item [6], the researchers have used wavelet transform for image feature extraction, whilst K-means algorithm has been used for classification (in the segmentation process). The crux behind the working of the proposed algorithm is to segment, extract, store, reduce and then classify and compare using a back propagation neural network (BPNN). The segmentation is done using K-means clustering; the important and significant areas of the leaf image are recognized during this process of the proposed algorithm. Once the image has been segmented, wavelet transform is applied to extract important features from the image. In this research item, a reduction process is performed using PCA to reduce the target image in size, as to speed up the computation.

3.3.1.1 Critical analysis of the research item

In this item, the researchers were able to get 98% accuracy using the BPNN classification technique. This research study particularly stands out in implementation of traditional image processing techniques, as it tries to reduce the extracted features using PCA. This certainly has an impact on the overall efficiency of the proposed solution and is a point in the favour.

3.3.1.2 Relationship to the proposed research work

This item elaborates on one of the many traditional image processing techniques proposed to detect disease ridden cotton plants.

3.3.2 Cotton Leaf Disease Detection & Classification using Multi SVM

In this research item, a different technique has been applied. In [6] a back propagation neural network was used to classify the images, however in this item, SVM has been used to classify the images. Albeit similar steps have been used up to the classification stage (with some modifications or changes). Image segmentation is a similar theme across most research papers as it helps in differentiating the background from the target image part (the one that needs to be compared), this helps in reducing useless comparison and helps in focusing the proposed algorithm or workflow to the part that matters, i.e., the leaf and its structure. The extractor in this study focuses on the color and texture features, so they are analyzed and extracted. In texture feature extraction, Color-co-occurrence method is used to quantize the image. Later, this quantized image is converted to GLCM, from whereon, the texture feature values are extracted. Lastly, classification in this study is done using the Support Vector Machines (SVMs) algorithm.

3.3.2.1 Critical analysis of the research item

In this item, a different approach towards classification and extraction is used. The proposed solution endeavours to use GLCM for feature extraction whilst using SVMs to classify the extracted image data. The results of the recognition activity are certainly encouraging. With a total of 87.5% accuracy, the proposed solution was able to recognise strongly (with a 100% accuracy) Red Spot Diseases and Healthy Leaves. However, the proposed solution was only able to correctly recognize Crumple Disease 60% of the times. According to the NIH website, the crumple disease is endemic in Pakistan (the target country for the proposed research work), which would certainly raise the chances of the prospective deployment of the proposed solution.

3.3.2.2 Relationship to the proposed research work

The proposed solution in this study uses SVMs as the classifier method and a different method for image segmentation techniques.

3.3.3 Fungal Disease in Cotton Leaf Detection and Classification using Neural Networks and Support Vector Machine

This study has suggested the use of image processing techniques to identify a diseased cotton plant. However, the diseases targeted in this study are focused and relate to the fungal stratum of the diseases that harm a cotton crop or plant. There are as follows: ALSFD, GMCD and RFFD. In this research item, images of diseased leaves, exceeding the 3000 number mark, were used to correctly evaluate and generate the results quoted by the study. The proposed process follows a similar structure as proposed in [6], [7], with the classification and extraction process being the differentiating factor. The proposed solution in [8] makes use of both, ANNs and SVMs (used in [7]) to classify the extracted portions of the images. However, the proposed workflow in this study extracts features such as the Contrast, Correlation, Energy, Mean, Standard Deviation, Entropy, Variance and Kurtosis from the cotton leaf image. The research papers [6], [7] make use of different image processing techniques to segment and analyze the input image for the appropriate disease detection.

3.3.3.1 Critical analysis of the research item

This study focuses on the category of fungal diseases in cotton crops. It uses a set of parameters that differ from the studies in [7], [6] to extract relevant data. Alike [7], it also uses K-means clustering to segment image data. This study certainly stands out as it compares two classifiers, SVM and ANN. According to the data published as part of the study, the SVM classifier performed better when compared to the ANN classifier. This result would certainly open the prospect of trying the proposed solution and data in [7] with K-means clustering as the segmentation technique. The classification results for the SVM Classifier are shown below.

Table 6: Classification Results for SVM Classifier

This table shows the classification results of the SVM classifier over the 3 types of fungal diseases identified

SVM Classifier	ALSFD	GMCD	RFFD	Accuracy
ALSFD	1123	74	135	84.3%
GMCD	4	1012	9	98.7%
RFFD	32	45	1056	93.2%

3.3.3.2 Relationship to the proposed research work

This study proposes another imagery technique to use for detection of diseases in the cotton cash crop.

3.3.4 Identification of Cotton Leaf Lesions Using Deep Learning Techniques

In this study, the classifier was replaced with CNN. The main aim of this study was to compare the use of traditional image processing techniques with CNN [9] to identify prevalent lesions of cotton leaves. Furthermore, this paper is more targeted on lesion detection on the cotton leaf.

3.3.4.1 Critical analysis of the research item

This study is certainly important for the cause of this project. The main objective of the latter is to compare the CNN methods with the traditional image processing techniques. As the results show, CNNs were able to achieve results 25% higher than those obtained by traditional algorithms.

3.3.4.2 Relationship to the proposed research work

This study introduces deep learning techniques i.e. CNN for disease detection.

3.3.5 Using Deep Learning For Image-Based Plant Disease Detection

This study is much more of a general nature. It establishes the use of Deep Learning principles in image-based plant disease detection. It compares the use of GoogleNet and AlexNet as the selected CNN architecture. This study is more generalized in nature and does not focus on any one plant. It targeted a total of 14 crop species, using an image data of 54,306 images which were classified in 38 classes [10].

3.3.5.1 Critical analysis of the research item

The model trained in this study was able to achieve a top accuracy of 99.35%. Furthermore, the study clearly establishes that even though the training of a CNN model can take up significant time, the classification on the other hand is instant. Furthermore, inside the published results of the study, it has been stated that when target images are taken in varying environments (from training images), the accuracy of the suggested model is greatly reduced. Therefore, photo environment clearly has a role to play in this study. It suggests using a dataset with more coverage.

3.3.5.2 Relationship to the proposed research work

This study establishes the use of CNN for image-based plant disease detection. It furnishes results that are certainly helpful in perceiving the use of CNNs in the suggested field of work.

3.3.6 Deep Learning For Tomato Diseases: Classification And Symptoms Visualization

This research study targets tomato crop diseases however is imperative when coupled with its findings. It also uses similar CNN architectures as [11] and establishes the case for CNN over the traditional methods. Furthermore, it specifies that the study used Goodfellow, Bengio, Courville and PlantVillage.org to obtain the datasets that were used in the study. The study, although has only focused on the images of tomato leaves.

3.3.6.1 Critical analysis of the research item

This study has concluded that CNN are, effectively, the better classifiers when it comes to disease detection using leaves in tomato crops. It has compared also, the AlexNet and GoogleNet CNN models. Furthermore, the study has compared the deep learning models and traditional imagery techniques and has published the result data in a tabular format, shown below.

Table 7: Accuracy Difference Between Deep and Shallow Models

The following table shows the accuracy difference between deep and shallow models.

			Without pre-training	With pre-training
Deep models	AlexNet	Accuracy	97.354 ± 0.290	98.660 ± 0.123
		Macro precision	96.566 ± 0.388	98.005 ± 0.282
		Macro recall	96.266 ± 0.414	97.850 ± 0.383
		Macro F	96.368 ± 0.262	97.911 ± 0.120
	GoogleNet	Accuracy	97.711 ± 0.149	$\mathbf{99.185 \pm 0.169}$
		Macro precision	96.989 ± 0.506	$\mathbf{98.529 \pm 0.194}$
		Macro recall	96.783 ± 0.350	$\mathbf{98.532 \pm 0.490}$
		Macro F	96.582 ± 0.161	$\mathbf{98.518 \pm 0.191}$
Shallow models	SVM	Accuracy	94.538 ± 0.301	
		Macro precision	93.317 ± 0.752	
		Macro recall	92.917 ± 0.461	
		Macro F	93.067 ± 0.303	
	Random forrest	Accuracy	95.467 ± 0.004	
		Macro precision	94.628 ± 0.004	
		Macro recall	93.808 ± 0.006	
		Macro F	94.185 ± 0.005	

3.3.6.2 Relationship to the proposed research work

This study is imperative in establishing the case for the use of CNNs over traditional methods.

3.3.7 Literature Review Summary Table

Table 8: Imagery Techniques to Detect Plant-based Diseases

The various computer imagery techniques used to detect plant-based diseases are presented here.

No	Name, reference	Authors	Year	Techniques Used	Description
6.	Disease Detection of Cotton Leaves Using Advanced Image Processing, [6]	Chaudhari, Vivek, and C. Y. Patil	2014	BPNN, KNN, Traditional Imaging Technique	Reviewing traditional imagery techniques involving BPNN as the feedback system whilst using KNN inside the segmentation.
7.	Cotton Leaf Disease Detection & Classification using Multi SVM, [7]	Patki, Supriya S., and G. S. Sable	2016	Multi SVMs	This study used Multi-SVMs as the classifiers and established different parameters for extraction.
8.	Fungal Disease in Cotton Leaf Detection and	Kumari, Ch Usha, et al.	2019	Traditional techniques, multi SVMs	This work compares the use of ANN and SVMs whilst establishing a

	Classification using Neural Networks and Support Vector Machine, [8]				whole new array of parameters.
9.	Identification of Cotton Leaf Lesions Using Deep Learning Techniques, [9]	Caldeira, Rafael Faria, Wesley Esdras Santiago, and Barbara Teruel	2021	CNNs, GoogleNet, AlexNet, Traditional techniques	Establishes the use of CNNs for disease detection through images. Compares traditional models to deep learning models.
10.	Using Deep Learning For image-Based Plant Disease Detection, [10]	Mohanty, Sharada P., David P. Hughes, and Marcel Salathé	2016	CNNs, GoogleNet, AlexNet, Traditional techniques	Compares AlexNet and GoogleNet performances alongside the traditional techniques.
11.	Deep Learning For Tomato Diseases: Classification and Symptoms Visualization, [11]	Brahimi, Mohammed, Kamel Boukhalfa, and Abdelouahab Moussaoui	2017	CNNs, GoogleNet, AlexNet	Showcases the accuracy boost when using CNNs for disease detection in the tomato plant. Uses tomato crop leaves for analysis.

3.3.8 Conclusion

A digital revolution in the agricultural field could be initiated using such simple and already researched techniques. A score of small-scale farmers live their lives one paycheck to another and must spend their savings when sowing the crop for a particular season. Enabling detection of diseases without incurring further costs and suggesting relevant mitigation techniques could prove to be of much help to the common farmer. Therefore, a number of techniques were reviewed as part of this project, these techniques were compared with one another in the hopes of finding the best solution.

3.4 Detailed Literature Review on Cotton Crop Growth Analysis

The idea to revolutionize agriculture using modern technology is now one of the biggest research topics in the contemporary world. The main objective to use these modern methods is to help the industry in management of different resources such as water, fertilizers, etc. The end goal is simple; to increase the yield of crops using modern technology. Many researchers have linked factors such as soil moisture, nutrient level, root depth and soil textures as some of the multiple factors which can affect the yield of a crop. However, some also noted that increase in soil moisture, nutrient level etc. increased the yield of the cotton crop. Our problem consists of analyzing the growth of the cotton crop and proposing solutions for better resource utilization whilst increasing the yield of the cotton crop.

3.4.1 A Study of Cotton Grown Under Constant Soil Moistures

Soil Moisture is one of the most important indicators which contribute to the yield of any crop: likewise, it is also very essential for a Cotton Crop to maintain its soil moisture levels. There are multiple ways in which this feat can be achieved: one is by simply relying on the rain cycles and the second is using irrigation methods. Water is an essential component of the cotton plant, where if the water content falls below the water scarce level, the plant can die. The research was carried out in controlled environments in pots of fixed sizes where their growth was observed after keeping certain parameters such as soil moisture constant.

3.4.1.1 Critical analysis of the research item

The strength of this research item in question is that it considered multiple factors and weighed them in relation to soil moisture. And what they observed was the fact that keeping the soil moisture constant resulted in better growth and yield.

3.4.1.2 Relationship to the proposed research work

As we intend to monitor the soil moisture level for growth analysis; we can draw the conclusion that maintaining a fair amount of soil moisture through-out the growth period of the crop will result in a better yield.

3.4.2 Recent Water Requirement of Cotton Crop in Pakistan

Recent years have shown that Pakistan's climate has been working on extremes where in some seasons there is no rainfall and in months of Monsoon there is ample amount of rainfall and the growing season for the cotton crop may coincide with rains which can be classified as torrential thus have the ability to damage the crop. In addition to this, since the rains are sporadic, most of the water flows out and is not harvested or can not be classified as contributing to the soil moisture. Thus, irrigation seems to be a viable solution to help maintain the soil moisture levels.

There are different calculations which can be done to determine the Crop-Water Requirement. Direct relationship between soil moisture levels and the yield of the crop has been established by many researchers and some have also classified the relation between rainfall and water requirement. For example, yield is affected when the crop receives rainfall which is less than one-fourth of the crop water requirement.

3.4.2.1 Critical analysis of the research item

The research item derives its methodology of calculating the water requirement of the cotton crop which can be used to ensure that water is not wasted, and the soil moisture is kept constant. However, the research item focuses more on assessing whether cotton crops in Pakistan can rely solely on rain-fed conditions and rules that it is essential for a secondary water source to be present.

3.4.2.2 Relationship to the proposed research work

This item concludes about the water requirements which are essential for the cotton crop. Weighing such quantifiers, we can quantify the water requirements and then process information to estimate the water cycles and durations for the crop.

3.4.3 Soil Site Suitability Evaluation for Cotton

Many factors contribute to the yield of the crop such as soil depth, soil texture and nutrient content. In addition to this, nutrient content and soil moisture are interdependent. And the cotton crop naturally grows its roots deep down in the soil. The increase in any of the four factors (soil depth, moisture, texture and nutrient content) increases the yield of the cotton crop. Furthermore, for better yield in cotton crops, the soil should contain at-least 100mm of moisture, any number below this can prove to be uneconomical for the yield.

3.4.3.1 Critical analysis of the research item

This research did not only weigh in one factor for growth analysis; it weighed in many others such as nutrient levels and the soil texture. It observed that change in any of these increased the yield. Also, it was able to quantify the level below which the yield would prove to be uneconomical.

3.4.3.2 Relationship to the proposed research work

Based on the grounds of this research we can set a bound below which the yield for the crop would decrease.

3.4.4 Combination of Sentinel-2 Satellite Images and Meteorological Data for Crop Water Requirements Estimation in Intensive Agriculture

The study embarked upon using multiple images acquired from the Sentinel-2 satellite and then processed these images to look for water flux and analyse the water content in this imagery. The crop coefficient-based estimation was used. These coefficients were acquired by processing the Sentinel-2 Imagery to produce 21 vegetation indices.

Furthermore, a lot of data was calculated using the data acquired from the meteorological departments to learn more about the climate and potentially rainy/dry days in the climate as the study is more focused upon water management. This data was then processed alongside with the data gathered from the Sentinel-2 Satellite. A model was also developed to predict water and irrigation management plans based upon the data.

3.4.4.1 Critical analysis of the research item

The research utilizes the technology on the Sentinel-2 Satellite to look for water content for the crop and then utilizes this information to support irrigation decisions.

3.4.4.2 Relationship to the proposed research work

We aim to use the Sentinel-2 satellite to gather our data that depicts the soil moisture content in the crop and then use that information to help with water usage and yield.

3.4.5 A Crop Monitoring System Based on Wireless Sensor Network

The researchers monitor the growth indicators such as soil moisture, temperature and atmosphere by utilizing the IOT devices and setting up a wireless sensor network. For this, the devices were powered with one or multiple power sources backed by either DC batteries and/or solar power. A central processing unit was set up that would process the data acquired through the sensors. The sensors transmitted their information based on a timer.

3.4.5.1 Critical analysis of the research item

Setting up a wireless sensor network enables to get real time information of the crop thus efficiently monitoring the factors essential for crop growth. However, this requires sensors either manufactured uniquely or using only certain types of sensors for data collection. Not only this, but this could also result in many technical challenges for the end user to set up.

3.4.5.2 Relationship to the proposed research work

As a possible source to gather information for growth analysis of the cotton crop.

3.4.6 Literature Review Summary Table

Table 9: Crop Growth Analysis using Satellite Imagery and Wireless Sensor Networks

The following table presents a list of sources used to complete this section of the literature review.

No	Name, Reference	Authors	Year	Techniques Used	Description
12.	A Study of Cotton Grown Under Constant Soil Moistures, [12]	P. S. SREENIVASAN,M.Sc. (AGRI.)	1949	Glaze earthenware pots 12" in height and 12" in diameter	Learning about constant soil moistures could increase the yield of the crop.
13.	Recent Water Requirement of Cotton Crop in Pakistan, [13]	Ghazala Naheed, Ghulam Rasool	2010	FAO Penman-Monteith Equation	Calculating water requirement for cotton crop
14.	Soil Site Suitability Evaluation for Cotton, [14]	J.L. SEHGAL	1991		Effects of changing Soil Moisture, nutrients etc on cotton yield.
15.	Response of Growth Stages of Cotton Varieties to Moisture Stress, [15]	Rajinder Kaur, O.S Singh	1991	Earthen pots (35 cm x 25 em) were filled with 1 : 1 soil and farm yard manure	Stressing Cotton when it is (6-7) weeks into it's growing period can help increase yield.
16.	Combination of Sentinel-2 Satellite Images and Meteorological Data for Crop Water Requirements Estimation in Intensive Agriculture, [16]	Jaouad El Hachimi, Abderrazak El Harti, Rachid Lhissou, Jamal-Eddine Ouzemou, Mohcine Chakouri, Amine Jellouli	2022	The FAO-56 model was used to estimate water requirements, Images of NDVI slice	Using slices of NDVI images taken at different intervals and then processing them to calculate coefficients such as Kc.

17.	A Crop Monitoring System Based on Wireless Sensor Network, [17]	Zhao Liqiang, Yin Shouyi, Liu Leibo, Zhang Zhen, Wei Shaojun	2011	Data Processing Unit, RF Modules, usage of batteries for power supply	Establishing Wireless Sensor Networks to real time monitor the soil data

3.4.7 Conclusion

Cotton is one of the biggest cash crops of Pakistan. Pakistan is aiming to become an industrial based economy and to achieve this feat it looks to focus heavily on the textile sector of Pakistan. The textile sector is completely dependent upon the quality supply of raw cotton. The yield of cotton crops can be increased by using modern methods to analyze the growth of cotton. By monitoring the soil moisture levels and keeping it constant the yield of the cotton crop can be increased. This has been concluded in multiple research aforementioned [12], [14], [15]. Furthermore, to achieve these feat multiple technologies can be used such as gathering and processing the images acquired by the Sentinel-2 satellite [16] or using wireless sensor networks to transmit real-time information which can be processed to make estimates regarding the water requirements [17]. However, using Wireless Sensor Networks can add to the cost of growing the crop. Not only that, but the end-user can also face many technical difficulties whilst setting up a wireless sensor network; and given the technical knowledge that is prevalent in our country today it seems not to be the best solution.

Chapter 4: Software Requirement Specification

This chapter lists down the software requirement specifications of the program to be produced as a result of this project. The operational aspects of the application have been highlighted in this document.

4.1 List of Features

1. **User Login**
    - Users of this application should be able to login using their assigned username and selected password.
2. **User Registration**
    - Prospective or new users should be able to register with the application as new users.
3. **Manage Plantation**
    - Users should be able to manage their plantations. Each user can have multiple plantations. This feature has been introduced to enable users to add plantations that are not contiguous are in different places geographically (separated plantations). The management of the said plantations will include the ability to add, delete or edit the said plantation. The edit ability will enable the user to rename the given name to the plantation, along with the approximate area that it covers.
4. **Analyze Crop Image**
    - This feature will allow the user to upload and analyze the image belonging to the chosen plantation. The respective image once uploaded, will be fed to an already trained Convolutional Neural Network model, which will diagnose a disease (if present) or return healthy status otherwise. In case a captured image indicates towards being disease ridden, this feature will tag the image with the appropriate mitigation technique (reverse matched with the returned disease ID).
5. **Monitor Crop**
    - The crop monitoring feature will allow the user to place a monitor (optional) on their plantation of choice. Once the monitor has been set (enabled), the user will be able to analyze and see the historical and contemporary weather patterns of the area. Furthermore, this feature will enable the user to estimate the amount of water the crop will need to maintain a healthy water cycle (dependent on the soil moisture) through weather trends and information extracted from the Sentinel Copernicus-2 satellite.
6. **Satellite Data Usage**
    - The application will utilize soil moisture data generated by the Copernicus-2 satellite to monitor soil moisture levels.
7. **Weather Pattern Usage**
    - A model trained using the historical and contemporary weather patterns can be used to improve crop inundation cycle of the cotton crop.

8. **User-friendly UI Design**
    - The application will present the user with an easy-to-use user interface that is inherently user-friendly. This will enable the user to encounter less complications and consequently provide a satisfactory experience.
9. **General Cotton Plantation Guidelines**
    - The application will present the user with a list of guidelines they can pursue to achieve a better cotton yield. These guidelines will include suggestions on before, during and after plantation stages. Moreover, data extracted from farmer surveys can also be presented to suggest useful pesticides and fertilizers.
10. **Already Trained Backend CNN**
    - Another useful feature will be the utilization of an already trained CNN in the backend. This network can be hosted on a cloud service in order to cut processing costs and improve return time.
11. **Real-Time Disease & Mitigation Techniques**
    - Since the disease and relevant mitigation techniques will be stored in central database (and then copied to the local storage), this will enable the user to experience a virtually real-time suggestion system. This feature will allow the application mitigation techniques to always be in sync with new research.
12. **Admin Managed Database & Mitigation Repository**
    - Admins (application maintainers) will be allowed to manage the existing disease names, mitigations and corrective measures.

4.2 Functional Requirements

1. User should be able to login once a valid username and password is provided.
2. New users should be able to register with the application as new users.
3. Each user can have multiple plantations.
4. User can add multiple plantations in different place geographically.
5. The management of the said plantations will include the ability to add, delete or edit the said plantation.
6. User can edit the plantation name and the area that it covers.
7. User can upload and analyze the image belonging to the chosen plantation.
8. Once the monitor option has been enabled, the user will be able to analyze and view the historical and contemporary weather patterns of the area.
9. On the dashboard, users can click on any list item of the recycler view that will redirect to the plantation data of that specific plantation.
10. If the user needs to check for a disease in a plantation, they will select the option for disease detection on the plantation page and will be able to select an image of the plant or use the mobile camera to take its picture.
11. As soon as a disease is detected for a plantation, its details must be added in that plantation’s history.

4.3 Quality Attributes

1. **Security**
   - It shall be ensured that the login or register page does not accept dummy input that tricks the database and eventually provides unauthorized access to the user to guarantee proper security.
   - The login and registration data must be encrypted before being stored onto the database.
2. **Availability**
   - Our application should always be readily available for use 24/7. It should be ensured that the server used by the app is always active.
   - User must be able to view the history regarding his plantation at any time.
3. **Compatibility**
   - App should be accessible by majority versions of android to ensure that all users of android are able to use it.
4. **Usability**
   - The UI/UX design of the application should be user friendly to ensure that the app's functionality is easily understandable by the users and that the layout changed for different orientations should be dynamic.
5. **Reliability**
   - It must be ensured that the disease detection process is highly accurate so that only the relevant mitigation actions are to be taken.
   - The weather forecast and monitoring data to be provided must be highly updated and accurate.

4.4 Non-Functional Requirements

- The system shall not take more than 2 seconds to login.
- The disease must be detected in less than a minute after providing the picture.
- The mitigation techniques must be displayed within 20 seconds after clicking the view mitigation techniques button.
- Soil moisture report must be provided in less than 30 seconds after providing the correct location.
- It shall be ensured that the login or register page does not accept dummy input that tricks the database and eventually provides unauthorized access to the user to guarantee proper security.
- Multiple users can access the app at a time performing multiple app functionalities.

4.5 Assumptions

1. The availability of internet with the user.
2. Possession of a mobile phone that can run the application in a satisfactory manner.

3. The user is technologically literate to a level where they can themselves navigate through the application.
4. The camera being used by the user to photograph a crop plant is greater or equal to **4MP.**
5. The camera in use is not technically defected and can take clear photos.
6. The user snaps a photo in daylight and not in nighttime using the camera flash.
7. The user can position the camera at the correct area and therefore include minimal background.
8. The internet is reliable and not very slow, as the image needs to be uploaded before it can be analyzed and processed.

4.6 Hardware and Software Requirements

4.6.1 Hardware Requirements

The following items will be required to run the application:

- An android phone in working condition with at least 4GB RAM.
- A router with reliable internet service.
- A working phone camera with a reliable camera (at least 4MP).

4.6.2 Software Requirements

- Working Android OS.
- Android OS which is a non-deprecated version.
- Correctly working onboard software involving camera and hardware drivers.

4.7 Use Cases

4.7.1 Login Use Case

Actors	App Users, Admin
Summary	The user shall provide their email and password on the login form and after successful verification, redirect the user to the home page.
Pre-Conditions	The user record must be in the database records added through registration procedure.
Post-Conditions	The user’s session is successfully established and shall be redirected to the home page.
Special Requirements	None
Basic Flow	

Actor Action		System Response	
1	The user opens the login page.	2	The login page is displayed asking for email and password.
3	The user enters a valid email and password.	4	The system verifies the email and password, establishes a session for the user and redirects the user to the main page.
Alternative Flow			
3	The user enters invalid email or password.	4	The system responds with an error message: Incorrect email or password entered.

4.7.2 Registration Use Case

Actors	App Users, Admin		
Summary	The user shall provide their email and password on the Register form and after successful verification, redirect the user to the main page.		
Pre-Conditions	The user record must not be in the database records added through the registration procedure before.		
Post-Conditions	The user's record is successfully added to the database and the user shall be redirected to the home page.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The user opens the register page.	2	The register page is displayed asking for email, password and retype password.
3	The user enters a valid email and password and retypes the same password in the retype password field.	4	The system verifies the email and matches password with retype password field, then establishes a session for the user and redirects the user to the main page.
Alternative Flow			

3	The user enters invalid email format.	4	The system responds with an error message: Incorrect email format, please enter a valid email.
3	The user enters an email that is already used for another account that has its record in the database.	4	The system responds with an error message: This email is already used by another user, please enter another email.
3	User enters different text in the password and retype password field.	4	The system responds with an error message: Passwords do not match, please enter the same password.

4.7.3 Disease Detection Use Case

Actors	App Users		
Summary	The user shall provide the picture of the cotton crop leaf by taking a picture through the phone's camera or selecting it from the gallery. The system will then match the disease with the on the basis of its training and after successful verification, redirect the user to the result page to display the disease details,		
Pre-Conditions	The system must already have been trained before to detect a variety of diseases with accuracy.		
Post-Conditions	The system must then identify the relevant mitigation techniques according to the type of disease identified it...		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The user opens the disease detection page.	2	The disease detection page is displayed asking for a picture through camera or gallery.
3	The user provides the image and the system starts matching the image with its previously trained disease detection system.	4	The system verifies the disease type and redirects to the results page to display the disease data there..
Alternative Flow			
3	The user provides a blur	4	The system responds with an error

	image and the trained system is unable to match it.		message: Image is unclear. Please retake the picture or select any other picture.
3	The user provides an image that is clear but does not show the leaf or shows any other plant part.	4	The system responds with an error message: The image does not show the leaf. Please provide another picture.
3	The user provides an image that is clear and shows the leaf. The trained system is unable to match it as it is not trained to detect that disease.	4	The system responds with an error message: Cannot determine the disease!

4.7.4 Mitigation Use Case

Actors	App Users		
Summary	The user shall click on the view mitigation techniques button that is present on the result page. After clicking that, redirect the user to the mitigation techniques page to display the relevant mitigation techniques according to the disease detected.		
Pre-Conditions	The system must already have been trained identify the relevant mitigation techniques for the disease detected. The view mitigation techniques button will only be shown once the disease is identified by the system.		
Post-Conditions	The system displays the relevant mitigation techniques according to the type of disease identified on the mitigation techniques page.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The user clicks on the view mitigation techniques button.	2	The system redirects to the mitigation techniques page and displays the mitigation techniques according to the disease identified earlier.
Alternative Flow			
1	The user clicks on the view mitigation techniques	2	The system redirects to the mitigation techniques page but is

	button.		unable to find the mitigation techniques according to the disease identified earlier so it displays the message: Unable to find the relevant mitigation techniques.

4.7.5 Monitoring Use Case

Actors	App Users		
Summary	The user shall provide with the location of their field and the system in return provides them with the soil moisture readings of their field.		
Pre-Conditions	The system must be linked to the satellite system that detects the soil moisture.		
Post-Conditions	The soil moisture report is successfully received, and the user is redirected to the soil moisture report page.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The user opens the monitoring page.	2	The monitoring page is displayed asking for location of the field either through entering the coordinates or by enabling location services.
3	The user provides the correct location through either entering the coordinates or through location services.	4	The system redirects to the soil moisture page to display the soil moisture report.
Alternative Flow			
3	The user selects to enter coordinates but does not enter them completely or in the required format.	4	The system responds with an error message: Incorrect coordinates. Please enter proper coordinates.
3	The user selects to provide current location but the loc. service is not turned on.	4	The system responds with an error message: Please turn on the location services.

4.8 Graphical User Interface

Figure 1: Startup Page UI
Figure showing the startup page.

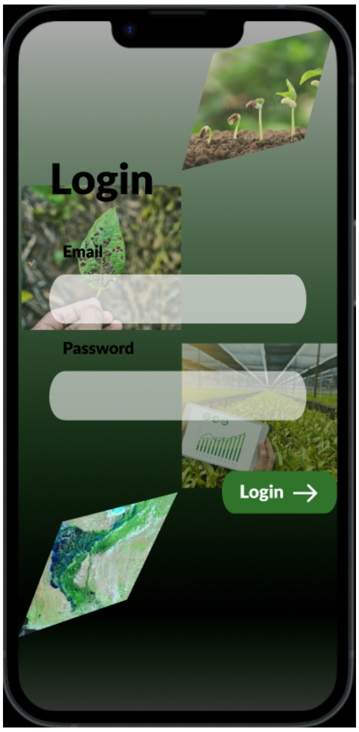

Figure 2: Login Screen UI
Figure showing the login page.

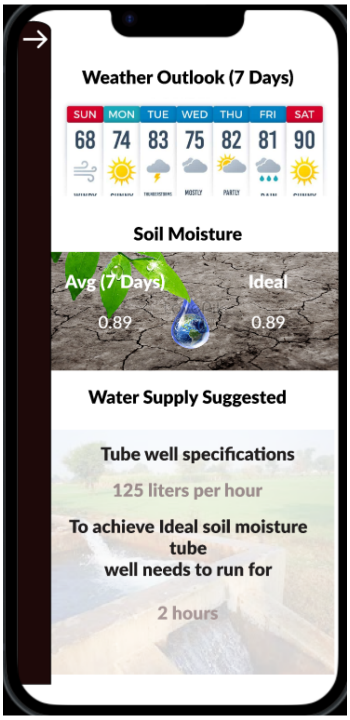

Figure 3: Plantation Management UI

Figure showing the plantation management UI.

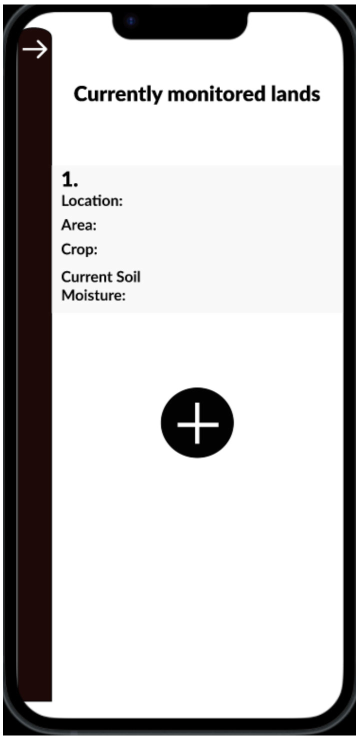

Figure 4: Plantation Monitoring UI

Figure showing the plantation monitoring page.

4.9 Database Design

4.9.1 ER Diagram

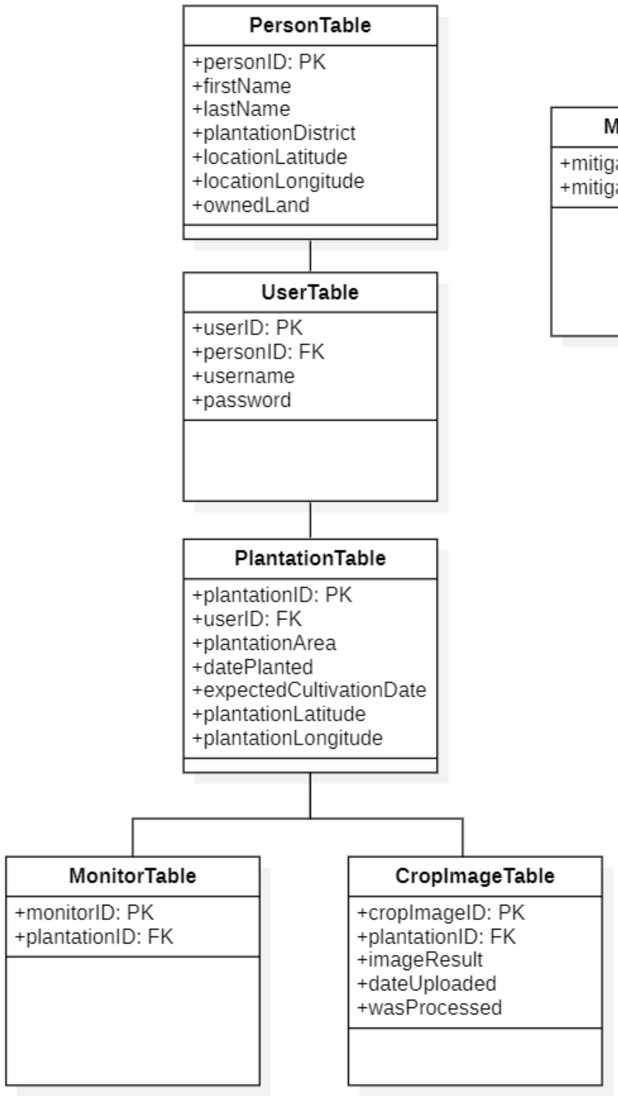

Figure 5: ER Diagram

Figure showing the ER Diagram that is to be used in the software.

4.9.2 Data Dictionary

Table 10: Data Dictionary for Person

The following table represents the data dictionary for the Person Table.

Field Name	Data Type	Constraints	Description
personID	INTEGER	**PRIMARY KEY**	Uniquely identified a person.
firstName	VARCHAR	**NOT NULL**	The first name of a person.

lastName	VARCHAR	**NOT NULL**	The last name of a person.
plantationDistrict	VARCHAR	**NOT NULL**	District where the plantation is.
locationLatitutde	NUMERIC	**NOT NULL**	Person current location Lat.
locationLongitude	NUMERIC	**NOT NULL**	Person current location Longitude.
ownedLand	INT	**NOT NULL**	Amount of owned land (in kanals)

Table 11: Data Dictionary for User

The following table represents the data dictionary for the User Table.

Field Name	Data Type	Constraints	Description
userID	INTEGER	**PRIMARY KEY**	Uniquely identifies a user.
personID	INTEGER	**FOREIGN_KEY (PersonTable)**	Links the user with a person record.
username	VARCHAR	**NOT NULL**	Username allotted.
password	VARCHAR	**NOT NULL**	Password chosen.

Table 12: Data Dictionary for Plantation

The following table represents the data dictionary for the Plantation Table.

Field Name	Data Type	Constraints	Description
plantationID	INTEGER	**PRIMARY KEY**	Uniquely identifies a plantation
userID	INTEGER	**FOREIGN_KEY(UserTable)**	References the userID against the plantation.
plantationArea	INTEGER	**NOT NULL**	Total plantation area.
datePlanted	DATE	**NOT NULL**	Date when the plantation was sown.
expectedCultivationDate	DATE	**NOT NULL**	Estimated date by the user.
plantationLongitude	NUMERIC	**NOT NULL**	Plantation location longitude.
plantationLatitutde	NUMERIC	**NOT NULL**	Plantation location latitude.

Table 13: Data Dictionary for Monitor

The following table represents the data dictionary for the Monitor Table.

Field Name	Data Type	Constraints	Description
monitorID	INTEGER	**PRIMARY KEY**	Uniquely identifies an active monitor.
plantationID	INTEGER	**FOREIGN_KEY(PlantationTable)**	References the monitor against a plantation.

Table 14: Data Dictionary for Image Table

The following table represents the data dictionary for the Crop Image Table.

Field Name	Data Type	Constraints	Description
cropImageID	INTEGER	**PRIMARY KEY**	Uniquely identifies the picture.
plantationID	INTEGER	**FOREIGN_KEY(PlantationTable)**	References a image against a plantation.
imagePath	VARCHAR	**NOT NULL**	Image Path that defines where the image is uploaded.
dateUploaded	DATE	**NOT NULL**	Defines the date time of upload.
wasProcessed	BOOLEAN	**NONE**	If the image was processed successfully by the CNN.
imageResult	INTEGER	**NONE**	Stores the result disease ID.

Table 15: Data Dictionary for Disease Table

The following table represents the data dictionary for the Disease Table.

Field Name	Data Type	Constraints	Description
diseaseID	INTEGER	**PRIMARY KEY**	The unique ID of the cotton disease.

diseaseName	VARCHAR	**NOT NULL**	The name of the disease.
diseaseType	VARCHAR	**NOT NULL**	Disease type. E.g. Fungal, Viral etc.

Table 16: Data Dictionary for Mitigation Table
The following table represents the data dictionary for the Mitigation Table.

Field Name	Data Type	Constraints	Description
mitigationID	INTEGER	**PRIMARY KEY**	Uniquely identifies a mitigation technique.
mitigationTechnique	VARCHAR	**NOT NULL**	The technique used.

Table 17: Data Dictionary for Disease Mitigation Table
The following table represents the data dictionary for the Disease Mitigation Table.

Field Name	Data Type	Constraints	Description
diseaseMitigationID	INTEGER	**PRIMARY KEY**	Used to identify a disease-mitigation record.
diseaseID	**INTEGER**	**FOREIGN_KEY(DiseaseTable)**	References the disease against a diseaseMitigation record.
mitigationID	**INTEGER**	**FOREIGN_KEY(MitigationTable)**	References the Mitigation against a diseaseMitigation record.

4.10 Risk Analysis

There are always some types of risk associated with any project, likewise, our project is not immune to such risks. The first and foremost risk is that our model fails to classify the picture; this can develop mistrust between the user and our application. Whilst there is no way to predict how and when this happens; still the failed image can be put inside a dataset of failed images and then the team could investigate such issues.

Furthermore, the second and by far the most dangerous risk that's associated with this type of project is that the model wrongly classifies an image and suggests the wrong measures. This can lead to wastage of user resources and possibly causing further complications for the user. Thus, it is highly pertinent here to let the user know that our model gives a suggestion, and it should not be taken as an answer from a specialized entity.

Moving on, since the nature of the product is heavily reliant on web architecture and systems that've been deployed on the web server, inadequate network speeds or poor connection can lead to the system failing and user not getting any response from the webserver. This issue is one which cannot be foreseen. In addition to this, since our user will be uploading an image using their mobile phones onto the webserver, we will need to use efficient compression algorithms to make sure that the information to upload is as minimal as possible to achieve efficient results.

The important thing to note here is that the application in its nature is heavily reliant on the web as this is where the database will be hosted and thus the risks associated with a web server failing are very much associated with such type of applications.

Chapter 5: Proposed Research and Methodology

The methodologies involved in completing this project are various. One of the methodologies pertains to developing and deploying the CNN model on the web server, the other pertains to developing the host application and the last is relevant to the setting up the central database server. Furthermore, another important methodology that needs to be covered pertains to the dataset that has been collected for training and testing the main model.

5.1 Dataset Collection

For this project, the dataset is a very important aspect. A convolutional neural network is unlike other neural networks and relies on itself for pattern recognition by using filters. Therefore, an important demand in this project was to collect a dataset that was ample in size and nature. For this, we have identified several sources.

The dataset, being photos of cotton plant leaves, must be compiled from various sources. The first source identified by us for this purpose includes user contributed repositories on Kaggle, GitHub and Mendeley Data. We are currently in the process of further identifying similar sources of importance. Several such sources will be used, and the images may need to be classified according to the disease. For this project, data augmentation is an important consideration. Since the conditions of when the photo is taken might differ, data augmentation can help in diversifying the image conditions by applying the relevant color filters on the image. Moreover, the dataset will have to manually loaded into the program and then used to train the model. One option could be to host the images on the same server and storage drive. In the training data folder, the images will be organized in directories pertaining to the disease name they reflect. The program would load the images and train the model. Once the model is trained, the testing images, stored in a separate directory will be used to query the model and get the result. To efficiently test the accuracy, the test data itself will have to be divided in the same categories (folders). A training-test split of 80-20 will be used.

5.2 Working Method

With the dataset covered, the second important aspect of the software involves the working of the software. The software will be catering to the need of the user, who is assumed to be the farmer who wants to use the application to better his cotton crop output. The farmer will have to register and then login to the system. Upon login, the farmer will have ability to manage their plantations. The relationship that exists between a farmer, the plantation and other derived entities is crucial to understand before looking at the overall algorithm at play. The user primarily owns land, which constitutes the location for a plantation. This land is referred in this document and the project as a **Plantation**. A user can have multiple plantations, which can be geographically distant and/or separated. The idea of the application is to be able to detect if a cotton crop plant (a singular plant from a plantation) is diseased. Therefore, a relationship exists between the Plantation in question and the plants that exist within it. The crop image, regardless of the result computed against it must be stored in the relevant plantation section to make tracking plantation progress possible. Obviously, a plantation has multiple other functions and parameters, but they will be explored in the relevant subsystem architecture section. Furthermore, another key feature of the system is to be able to monitor the plant growth. Therefore, a monitor, singular in nature can be associated with a target plantation. If a user has more than one plantation, they can have one monitor for each plantation, but no plantation can have more than one monitor. With that being explained, the diagnosis is done on a crop image, but the result is stored with relevance to the image and by consequence the plantation to which the image belongs. Once the result has been received by the central system, it processes that result and displays the result and the appropriate mitigation action to the user if disease was found, otherwise a confirmation prompt is displayed to the user. This constitutes the disease detection and the first part of the mitigation lifecycle. In case disease is observed, the disease mitigation lifecycle is triggered, and mitigation actions are suggested with the plantation being marked as diseased. Furthermore, the plantation section will be able to track all statuses. As to when the last photo was added, if the disease was diagnosed and what mitigation action was suggested. The specifics will be explained in more detail when the relevant subunit is addressed. Moreover, the same is true for the disease identification lifecycle.

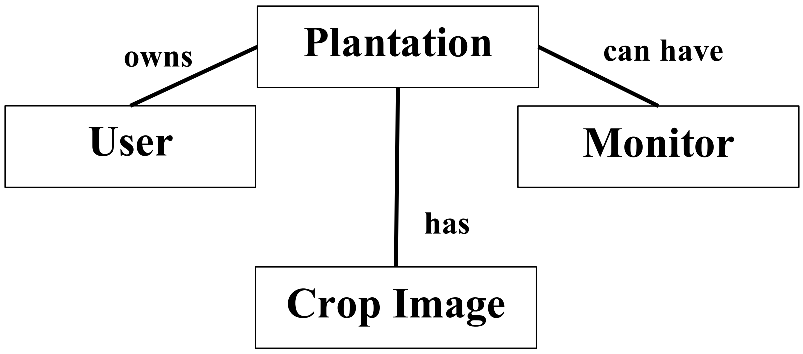

Figure 6: Basic Functional Relationships
This figure describes pictorially some of the important relationships in use in the program logic.

With the relationship established, the working of the software can now be looked at. The software is broadly divided into three subunits: detection, mitigation and monitoring. The detection part consists of capturing the image and then sending it to the model for analysis. The mitigation process involves suggesting mitigation techniques once a disease has been diagnosed, whilst a monitor serves the purpose of tracking moisture level and providing suggestions based on both: satellite data and weather patterns.

The disease identification module enables the user to capture an image relevant to a plantation. Once that image is captured, it is uploaded to a file server to a unique folder partition (identifiable by the username and the ID that the server allots to a user). Once the upload is successful and completed, the path (where the image is kept) and the image is sent to the database server (in a queried form) for record creation. The query will return the ID of the record back to the disease identification module. From here on, the returned ID is passed to the web server. The web server downloads the image and loads it to the model as a test data. The analysis is run, and the result (an integer value) is returned to the calling module. Before the return is done, the web server updates the record. On return, the calling module examines the returned ID against defined checks and forwards it to the mitigation module. The mitigation module is responsible for reverse checking this ID (now the disease ID) against its dictionary maintained on the database server. If a match is found, it does two things: prompts the user with the appropriate mitigation technique) and returns to the disease detection a collection which contains the mitigation technique and information. The detection module would then mark the plantation as diseases and record the mitigation technique suggested. The UI would then also reflect this change by changing the plantation outline color.

Lastly, the monitoring module is isolated from the abovementioned units. It is a standalone module that is only in contact with external APIs and a predictor model. The predictor model, which can be on board the software package or part of the web server is helpful in estimating the amount of crop or field inundation required to maintain a reliable soil moisture level. To do this, the monitoring module extracts information from the satellite API and weather API. In this case, it uses a combination of rain prediction and already identified soil moisture levels to guide the farmer.

Chapter 6: High-Level and Low-Level Design

This chapter emphasizes on the high-level and the low-level design of the software to be produced.

6.1 System Overview

The real intention with this section involves truly describing the core functionality of the project. The brief explanation on the design principles to be used or deployed is something of secondary stature. In this project, the core areas which need to be addressed have been divided into eight broad categories. They are as follows:

1. Login/Registration
2. Main Dashboard
3. Plantation Management
4. Disease Identification
5. Disease Mitigation
6. Crop Monitoring
7. Parameter Management

Each of these categories will contain in themselves further divisions that try to break down the visualized solution into relevant manageable modules. Such an action will help in reducing the design complexity and will enable code planning easier.

6.1.1 Login/Registration

The Login and Registration modules allow the user to access or gain access to the system. For our project, it is imperative for us to record personal information as it might be useful in coordinating with the user in case of a system error. This error can include wrong estimation and prediction of a taken photograph or wrong monitoring data provided. In such a case, the application may use the contact detail of the user to push alert notifications. Furthermore, the usage of user data will help in linking the plantation to the specific user and the said plantation to the images and so on.

6.1.2 Main Dashboard

Once the user logs in, the main dashboard will show up. The dashboard will serve as the central place for plantation management and relevant insights. The dashboard will enable the user to add new plantation entities and add or delete the existing ones. Furthermore, insight regarding the said plantation can also be hinted as part of the dashboard. Moreover, real time information that can be beneficial in growing the crop yield can be incorporated as part of the dashboard. This information will include suggestions on measures to better the plantation environment and manage the crop during the growing stage.

6.1.3 Plantation Management

This section will allow the user to manage their plantations. Plantations are in effect geographically separated land entities. As a user may have several plantations, this application will endeavor to cater to that need. By providing the ability to keep record of each plantation separately, the user will be able to more efficiently manage and track the crops pertaining to a particular plantation entity.

6.1.4 Disease Identification

Disease identification involves analyzing the image, once taken, against an already trained network. The identification itself is based on three parts: capturing an image, running it against the model, returning the result.

1. **Capturing the image**:
    - Image capture is done natively by the android application. Once an image has been captured, it is sent to the user for a first-look review. Once the user confirms the image as satisfactory, it is then saved locally. Once the image has been saved, it is now sent to a webserver (that hosts the predictor algorithm and the model alongside other things) along with a plantation ID for reference. This plantation ID is the unique identifier of the plantation to which the captured image belongs.
2. **Running it against the model:**
    - Once the webserver receives the image from the client application. It will firstly upload that image on the central file repository. A filename is assigned to the image. This filename is usually an amalgamation of the plantation ID and a random string. Once the path of the newly uploaded file is finalized, an entry in the central database is made.
    - When the record keeping is finalized, the image is now ran against the trained CNN model. The result is usually an ID that corresponds to the type of disease encountered. If no disease is found, a negative integer is returned.
3. **Returning the result:**
    - Once the image has been analyzed, the detected disease or otherwise an OK status message will need to be returned.

Once the result has been obtained from the network deployed on the web server, the appropriate action is suggested. If no disease is found, the relevant prompt is then sent back to the user.

6.1.5 Disease Mitigation

Disease Mitigation is concerned with suggested the appropriate mitigations to the user once a disease has been diagnosed in a plant. To achieve this, a real-time disease dictionary that reverse matches the disease to the mitigation technique is required. Such a dictionary will be maintained by the application admins to ensure the availability of the best possible technique against a particular disease. The latter is covered in Parameter Management, which is an admin task.

6.1.6 Crop Monitoring

Crop Monitoring involves analyzing the moisture data which can be fetched from the images which will be periodically captured by the Sentinel-2 satellite; these images can be further processed based upon the parameters publicly available on Sentinel-2 documentation to estimate the soil moisture level.

1. **Selecting the area:**
    - The user will be required to turn their location services on and using that location the geometric coordinates for the user will be fetched. After this, the user will be prompted to enter approximate coverage area (owned land) as an

input; using these both parameters the area in focus can be increased and images for this certain area fetched.

2. **Estimating water consumption/supply in regard to weather trends:**
    - Based upon the soil moisture information gathered our model will be able to predict the amount of water supply needed to reach the optimum soil moisture level. Now there are two considerations whilst making this decision: first and foremost is the weather. The average amount of possible rainfall will be fetched and using this information, the suggestion will be prompted to the user. The suggestion will contain the amount of time that they should artificially supply water to their crop using a source such as a water pumps. Second consideration in this process is the water output of the water pumps; if the user knows the output of their water pump, then that will be used, in case that it is missing, we will use the average of the output data across our platform or if that is unavailable then a general average of water pumps used in Pakistan.

6.2 Design Considerations

As the application is not stand-alone and involves the usage of web applications and third-party APIs, there are a couple of issues that need to be addressed. The first and the foremost issue pertains to the disease identification lifecycle. As was explained in the System Overview and before, the proper working of the disease identification lifecycle involves the usage of a web server and the storage of user captured images. Moreover, the monitoring part of this application involves two things. The usage of available weather data of a geographical area and data extracted from satellite imagery and consequently queried from the satellite API. Consequently, these three are the pertinent issue that need be addressed before devising a solution. Moreover, performance is an important measure that needs to be considered when writing a solution that involves the aforementioned.

The issues that need to be addressed are as follows:

1. **Web Server**
    - The Web Server will be used to host the trained CNN model. There are a couple of questions that need to be addressed before the web server is made active. These questions or issues are as follows:
        - Which web server service is to be used?
        - Where is the nearest server the web server can deployed?
        - What is the performance measure of servers that are not geographically present in Pakistan?
        - How should the interfaces be planned on the application and on the web server?
        - Which framework to use to deploy the model on the web server?
2. **Storage Considerations**
    - A large part of the program involves storing the already captured user images to maintain record of the crop plant. The issues that need to be addressed are as follows:
        - How much storage space should the application rent online?

- Should the storage for a particular user be duped locally?
- Should the application restrict the user to a primitive number of pictures that can be stored on the central database? (To save storage costs)

6.2.1 Assumptions and Dependencies

This application is intended for use by Android users. It is solely being developed for the Android platform. With the previous statement established, following are the assumptions taken as part of the development:

1. The availability of internet with the user.
2. Possession of a mobile phone that can run the application in a satisfactory manner.
3. The user is technologically literate to a level where they can themselves navigate through the application.
4. The camera being used by the user to photograph a crop plant is greater or equal to **8MP.**
5. The camera in use is not technically defected and can take clear photos.
6. The user snaps a photo in daylight and not in nighttime using the camera flash.
7. The user can position the camera at the correct area and therefore include minimal background.
8. The internet is reliable and not very slow, as the image needs to be uploaded before it can be analyzed and processed.

The dependencies include:

1. The trained model, which is deployed on a web server.
2. The Copernicus-2 Satellite and its API that helps in querying the parameters and returns the moisture result.
3. A deployed prediction model on the web server for crop monitoring and soil moisture maintenance.

6.2.2 General Constraints

In this project, there are a couple of constrains that impact the performance and the accuracy of the software system. It is imperative to reiterate that this project is in nature of a ‘suggestive nature’ to the user. What the previous system entails is the nature of the software package to suggest and predict issues related to a certain thing (a cotton leaf). Furthermore, the monitoring part of this software is based on data provided and captured from a satellite, coupled with weather trends of the geographical area.

Keeping the above paragraph in mind, the general constraints of the software are listed as follows:

1. Since the disease identification lifecycle is based on a trained CNN, an important part of the project is training the CNN itself. The network is as efficient as the data provided.
2. The monitoring ability of the application is totally dependent on external data taken from satellite imagery and historical weather patterns extracted from other satellite missions.

3. The application requires a connection to the internet to work. This is because the model itself is hosted on a web server.
4. The application is required to be as performance efficient as possible.
5. Crop Images, pertaining to a specific user plantation may need to be primitive. As the images need to be stored on the central database to enable plantation management.

6.2.3 Goals and Guidelines

1. **Problem Partitioning**
    - The development should be done by partitioning the problem into sub problems and then using a bottom-up approach to write and design the software.
2. **Keep it Simple Stupid (KISS Principle)**
    - Whatever the solution, should be as simple as possible.
3. **Proper implementation of Object-Oriented Development Concepts**
    - The application design and proposed solution should implement good OOP concepts. The class relations should be well defined and the ambiguity should be as low as possible.
4. **Ensuring model accuracy**
    - The prediction models should be well trained so that the performed predictions are accurate to a high degree.
5. **Ensuring production grade software**
    - Another key goal is to ensure building a software that is production grade and professional.
6. **Professional UI outlook**
    - It is imperative to ensure that the UI outlook is professional and user friendly.

6.2.4 Development Methods

6.2.4.1 3 Layered Architecture:

Layers depict different abstractions and levels of a software. Layered architecture helps to divide the application into multiple segments to support numerous implementations. Every layer must on be able to access the layer beneath it. UI layer is the highest layer, it contains the software interface. It generally focuses on the logic of how the data is to be presented to the end user. Followed by this, the business layer contains most of the business logic of the application. It focuses on the core functionalities of the app. It can also add, delete of modify the data in the data layer. The data layer as the name suggests, has access to the database of the system. It has the access to read or write to a database All the data of the application is stored and managed here. It will have access to relational database system such as MySQL, PostgreSQL, Oracle and other systems.

We will be using this as it enables faster development by the fact that each layer can be worked upon simultaneously. Furthermore, it facilitates scalability as each layer can be scaled individually which also reflects its flexibility. Adding to this it also improves reliability and security as a fault in one layer is less likely to affect the performance and availability of other layers.

6.2.4.2 SOLID Principles

SOLID design principles are intended to make the software design easier to understand along with improving the scalability and maintainability of the system. An object's design is important in the software development process, especially in determining the accessibility scope. Solid principle provides easy access to manage the object entities. The SOLID principle integrates the feature of ensuring that the debugging process of software is much easier. The code also becomes relatively easier to read and understand.

6.2.4.3 Interfaces

The interface defines what can pass between the system and the environment. An interface is a description of the actions that an object can do. In terms of object-oriented programming, interfaces define the functions that an object can have. Unlike a class, an interface does not implement methods; instead, methods defined by the interface are implemented by the classes that implement the interface. A class can implement multiple interfaces. Interface allow us to take an overall overview of what the main functionalities will be related to. Use of interfaces enables abstraction and gives the flexibility to modify your program in the future. Interface makes unit testing easier**.** For instance, if your program is making use of interface for dependencies then you can easily replace those dependencies with a mock, stub, or a dummy for testing. Furthermore, interface allow faster development and increased flexibility. If you are coding as a team for a client-server application, use of interface will enable you to list down the main functionalities at first of each part which also improves the clarity and then implement them later on simultaneously without depending on the completion of one whole part while waiting for the other. Loosely coupled systems can be developed through interface. Polymorphic behavior can be implemented through them.

6.2.4.4 MVVM (Model View ViewModel)

MVVM proposes the separation of the core business logic part from the data presentation logic of the application. It makes use of separated code layers. Model layer manages the abstraction of the data sources. Model and ViewModel layer work together to save and retrieve the data. The main objective of View layer is to communicate the ViewModel layer about the user's action. This layer has no kind of application logic, it basically observes the view model. ViewModel is the layer between model and view. It acts as a link between the previous two layers. Only the data streams that are essential to View are exposed by this layer. MVVM is implemented through either the databinding library published by google or through RxJava tool for databinding. However, in our case the benefits of using MVVM are outweighed by its flaws so we will not be using this for the development of our project. MVVM can be an over kill for simple UI's. Consequently, in large scale cases, it becomes very difficult to design the ViewModel. On more major disadvantage of using MVVM is that it becomes difficult to debug because of the complex data bindings.

We will be using the first three methods for the development of our application due to their obvious benefits described above.

6.3 System Architecture

In this section, the high-level architecture of the software will be explored. Before diving deep into the architectural diagram and therefore the architecture of the software system. It is imperative to first look at the decompositions and the units that make up the high-level design of the system.

The software will primarily involve 4 units of major importance. The program dashboard, disease identification, disease mitigation and then the monitoring lifecycles. The program dashboard might seem as the odd one out, but this unit is specifically of importance as it helps in representing the true purpose of the application being developed. In order to correctly understand the need for the program dashboard as a major subunit, it is important to visualize the software in action. Firstly, the major actors in this software are twofold: the user, who is in turn a farmer and the system itself, which contains the trained neural network, disease and mitigation dictionaries. Moreover, it is important to understand the relationship that exists between the user and managed entities.

The dashboard is imperative for proper plantation management and tracking of crop health. The dashboard subunit of this project allows the user to correctly navigate through their existing plantations and add new ones if required. Furthermore, it provides an important start point for the navigation of the software to start. The workflow for the dashboard process is defined in the process model below:

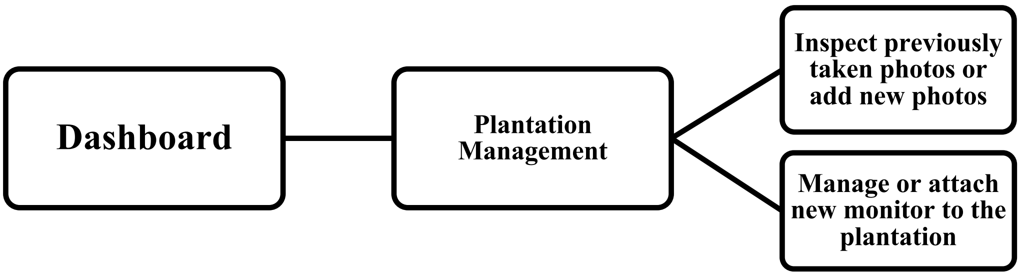

Figure 7: Dashboard Simplified
This figure shows the actions performed by the dashboard in a simplified manner and stresses on its importance.

To reiterate, the dashboard may contain more than one plantation for the user to choose from. As stated in the process model above, the user will be able to manage similar pages for each plantation entity. Obviously, the process model above has been kept short as the inspection and monitor management part itself has an independent workflow.

Another important consideration involves visualizing where the detection model is kept and how it is trained. Firstly, the model is hosted on a webserver and the application will communicate with the webserver and a central database to analyze the image on the model. More in depth commentary on the analysis and detection lifecycle is done in the relevant subsection. As for the model, the team is primarily responsible for the upkeep of the training and maintenance of the model. Initially, the project team will be responsible to ensure that the model is efficiently trained. The result, from the model can be only interpreted in accordance with the existing disease-mitigation dictionary present on the system. The upkeep of such a dictionary will also be an important part of the project. The upkeep and maintenance will require the development team to correctly document and store the techniques coupled with the associated diseases. For this, each disease can be assigned an ID. This ID would be generated by the CNN as a result of the analysis and can also be used to find the suggested mitigation technique.

The monitoring submodule of this application is concerned with reading and querying data from the relevant satellite and checking the Copernicus Satellite. Furthermore, a prediction algorithm or model is implemented as part of the same web server as the CNN (albeit different submodule) to perform soil moisture and inundation predictions. This prediction algorithm or model is mostly of a suggestive basis. What it includes and considers is the real time soil moisture data obtained from the satellite along with the weather predictions of a specified longitude and latitude for an arbitrarily decided time frame. The monitoring feature is open to expansion as it can further include analysis of the future weather data and can alert the user if the weather in future is hazardous to the crop health. Furthermore, this monitoring module can push notifications to the user in case the inundation cycle is abnormal, or some parameters are not performing as expected. This module will mostly rely on real time data gathered from the satellite and weather API.

With the subunits description covered, it is rather pertinent for us to move to the Architecture Drawing of this software system. The architecture diagram is a visual depiction of the software or the android application and its interaction with the different modules that make up the system. The communication can be visualized as below:

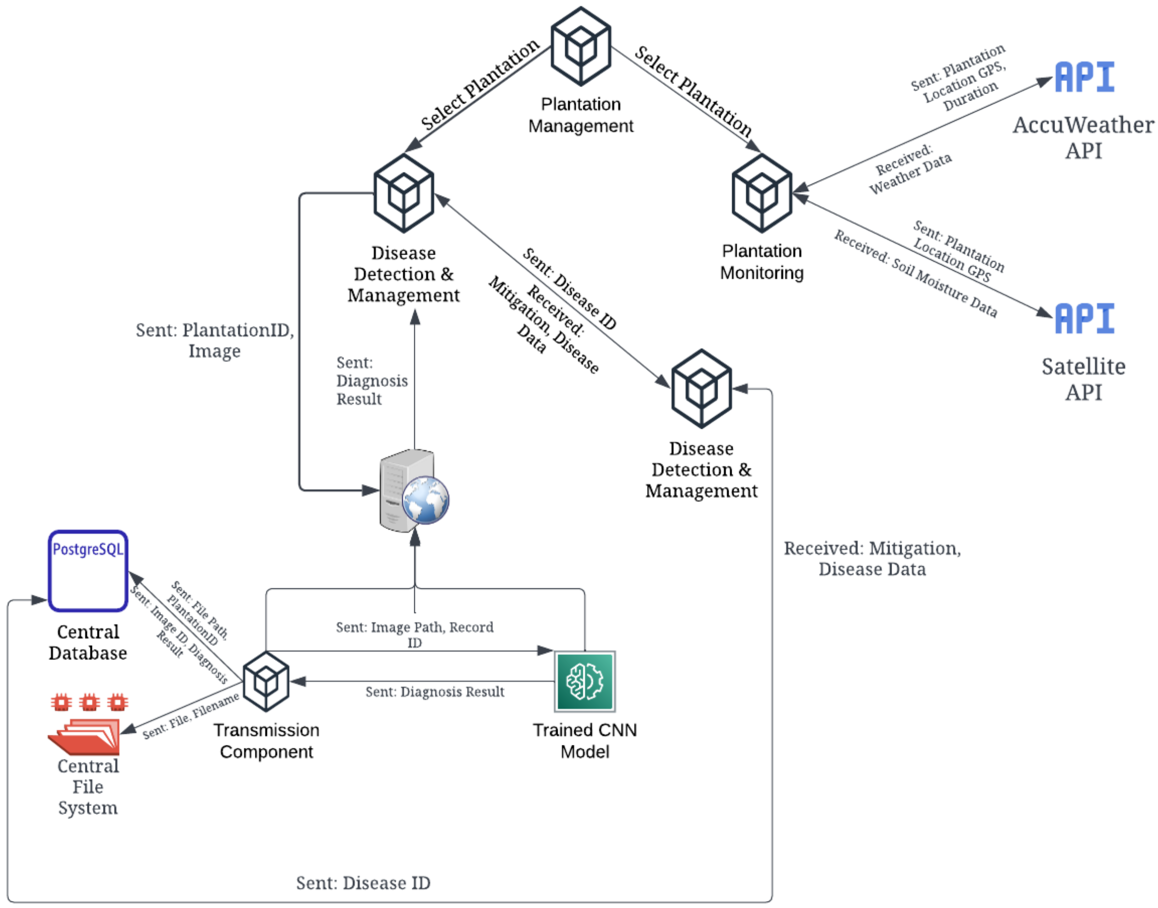

Figure 8: System Architecture Diagram

This diagram explains the system architecture of the software to be produced.

The main software is the software process, the main thread or routine that is communicating with other subunits. The main software, when signed into by the user will redirect to the plantation management dashboard page of the application. The Login-Register subunit has been skipped as it is a minor operation and does not represent the core functionality of the software. Once the user logs in, they are redirected to the application dashboard. The dashboard lists the user owned plantations, which can be navigated by the user. Once the user navigates to their plantation of choice, they can upload images to it using their phone camera. Once the user clicks a photo using their phone, they are prompted again to ensure that the image they have captured is satisfactory. If on review, the user proceeds with the recently captured image, the image is then sent to the webserver alongside the relevant plantation ID. This webserver hosts the CNN model for the disease detection subunit. Moreover, it also hosts the service which is responsible for receiving information from the client application and creating records on the file server and the database server against that received information. Once the receiving service receives the plantation ID and the image, it does two things. Firstly, it uploads the image to the central file server and assigns it a filename. This filename is usually a combination of the plantation ID and a randomly generated string. Once the upload is complete and the path to the new upload has been determined, this path is now passed to the CNN model. The CNN model loads the picture and runs it against itself. The result from the CNN model is usually an identifier that corresponds to a particular disease, if disease is detected. Otherwise, the result is a negative integer that signals that no disease is present. This ID is then returned to the client as the diagnosis result. Moreover, the web server updates the entry of the processed image ID with the recently computed diagnosis result.

When the client receives the diagnosis result (in form of an ID), a check is performed to see if the result is legit (disease was found). If the disease is confirmed, the ID is searched for in the disease-mitigation dictionary and the disease information combined with the associated mitigation action is saved.

Once the image data is processed to get real data for the soil moisture levels, then using this soil moisture level a twofold estimation can be made: firstly, the average amount of expected rainfall will be fetched from a reliable weather source and using this information we can predict how much water output we will need from artificial sources such as a water pump to achieve the optimum soil moisture level in the mapped area. This information will help our model to reduce the attached cost of operating a water pump for the user.

Most of the sub-architectures have been emphasized in their relevant sub-architecture heading.

6.3.1 Plantation Management Architecture

The Plantation Management submodule is linked with the Dashboard of the application. This is a particularly important subsystem as it controls the user experience and adds to the user ability to navigate through the features available. Firstly, each user can maintain and navigate through multiple plantations. Any plantation is essentially the land where the cotton plant has been sown. One user or farmer can have multiple geographically separated instances of land, therefore, the ability of a user to manage multiple plantations is implemented.

Secondly, for each plantation there are two possible actions: 1) monitor the plant, and 2) check the plant pictures for possible diseases. Therefore, the two are linked and owned by the plantation. Each plantation can have a monitor, which can only be singular in nature (this is handled by the crop monitoring architecture). Whilst each plantation can have any number of crop images taken by the user for disease analysis. These pictures are not common across plantations, every plantation will have pictures relevant to itself only. By storing the image after analysis and the relevant generated result, the health progression of the crop is documented. The analysis is done on the crop image and this analysis is handled by the Disease Detection Architecture.

Furthermore, this subsystem will help the user in tracking relevant statistics and maintain a history of the plant. It will also suggest to the user any active mitigation suggestions that have been generated as a result of disease detection. The system will mark any such plantation as diseased and record the time of determination and rely on user input to mark it as healthy again or ask the user for a rerun of the crop image. The mitigation suggestions are recorded by the mitigation subsystem, which is called by the disease detection module.

To reiterate, the purpose of this subsystem is multifold. It aspires to serve as the main controller by enabling navigation through many plantations, individually track the health of the mentioned plantations, remind the user of any active mitigations (relevant to a plantation) and enable the user to see monitor specifics (relevant to a plantation). The monitoring of a plantation is done by the crop monitoring subsystem. The diagram below aspires to graphically document the usage of this subsystem.

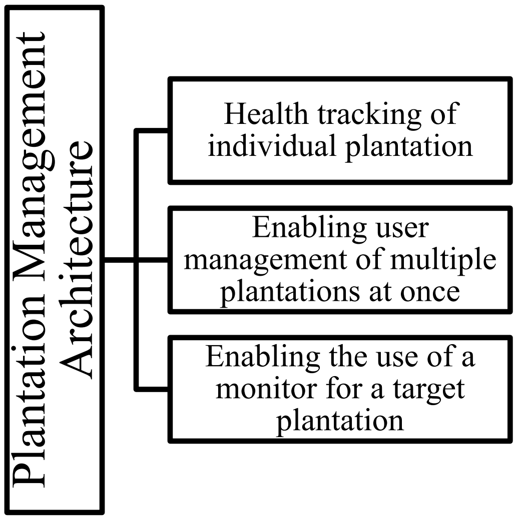

Figure 9: Plantation Management Subsystem
This figure puts in picture the actions carried out by the plantation management module.

6.3.2 Disease Detection Module

The Disease Detection subsystem is primarily twofold in nature. It is important to note what this subsystem receives from the program as a parameter. The picture, taken by the user is received as a parameter, along with the ID of the plantation the picture is for. There are two aspects to what this architecture is responsible for. Once the image has been captured, the image needs to be uploaded and an appropriate record needs to be made in the database. After this, the ID of the image (relevant to the central database) needs to be relayed to the CNN Model Web Server. Both have been described in the headings below.

6.3.2.1 Image Capture and Transmission to Web Server

This Central Database exists online as an independent entity. The reason why the database has been placed online relates to the CNN Model Web Server. Since the identifying model is kept online, there is no point in localizing the database. Plus, the same database is responsible for ensuring synchronization between the model and the software system. Once an image is received by the disease detection webserver, it is first uploaded to a central storage (file server). Once the image has been uploaded and stored, the image properties are then sent for storage in the central database. These properties include the image name, the path where the image is stored and the associated plantation ID as a reference. This subsystem, then passes that information to the central database and creates a record against the uploaded image. Each record has a unique ID, a reference to a plantation, the path of the image and the result associated with every image.

6.3.2.2 CNN Model

The CNN model receives the ID and the path of the image that was previously uploaded by another subunit on the same webserver. Once it receives the path, the image is loaded onto the model memory and then processed. When the model is done processing the image, the output can conform to one of the two things: disease not found, disease found (and the appropriate ID). The result computed by the CNN model is then sent back to the calling subunit and by consequence to the client application that initiated the det

The harmony of the Detection Architecture can be explained in the figure shown below.

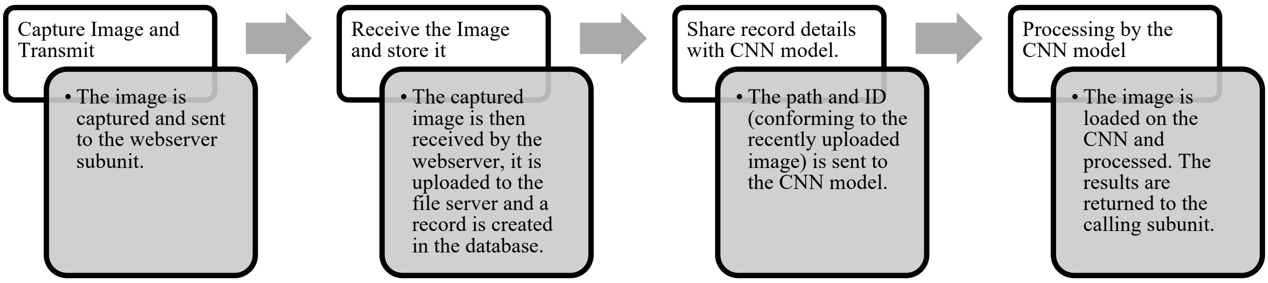

Figure 10: Disease Identification Subsystem
This figure explains the disease identification lifecycle.

6.3.3 Disease Mitigation Architecture

The disease mitigation architecture suggests mitigations once a disease has been diagnosed by the detection subsystem. As was stated above, the mitigation architecture receives a disease ID from the detection architecture (in case of presence), it then uses this ID to reverse search a mitigation technique and then suggests the mitigation to the user. In this process, several other things are done. This architecture will store the mitigation history relevant to a plantation and a crop image to ensure tracking of plantation health. This dictionary of diseases and relevant mitigation is maintained by the project team and is hosted as part of a centralized database server.

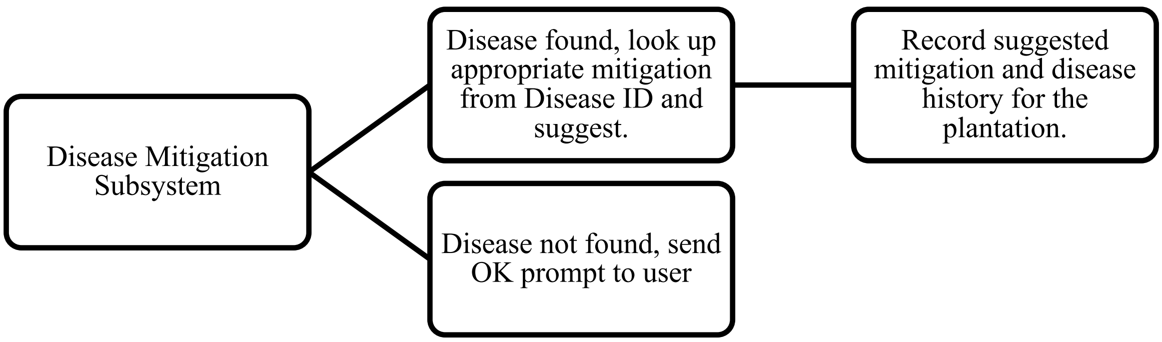

Figure 11: Disease Mitigation Subsystem
This figure explains the disease mitigation subsystem.

6.3.4 Growth Monitoring Architecture

The Growth Monitoring sub-problem can be architecturally divided as illustrated below. Using the data queried from Sentinel-2 satellite and the weather forecast over a period of days, our system should be able to make logical estimations regarding the amount of artificial water supply necessary to maintain the optimum level of soil moisture for crop growth. This will consequently help the user to minimize the cost associated with running their artificial water supply, thus, cutting the cost spent on the crop growth. The way that this architecture works is through harmony. It uses two external APIs to correctly suggest to the farmer the right path to take. Firstly, it uses the Satellite API to get the soil moisture data related to a particular GPS location (defined by the location latitude and longitude). Secondly, it uses the weather trends of the location to check for expected rainfall and the associated chances. Relevant to the chances and the weather outlook, it sends the information to the predictor model which suggests the appropriate action to the farmer. Furthermore, the weather trends system can be easily expanded to warn the user (or farmer) about expected hazardous weather and the precautions to take. Moreover, notifications can be pushed by this subsystem to warn the user of any abnormalities or suggest actions.

The diagram below explains the inundation suggestion system:

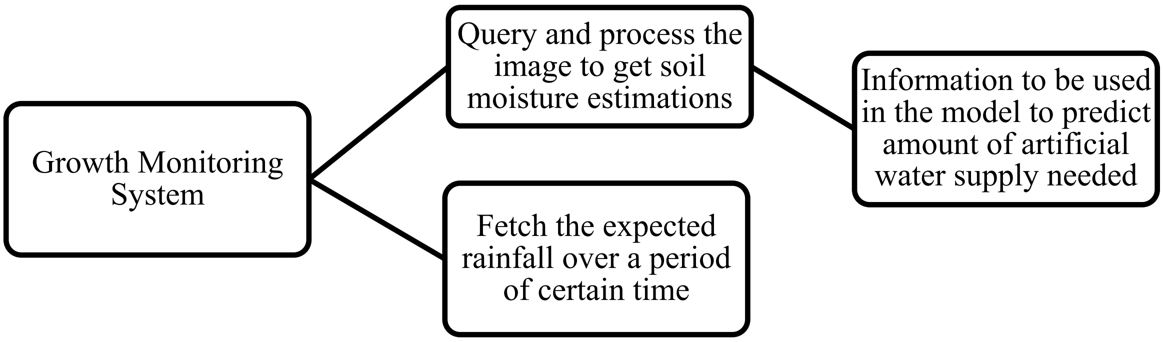

Figure 12: Growth Monitoring Subsystem

This figure puts into a diagram the monitoring actions taken.

6.4 Architectural Strategies

The purpose here will be to present the programming languages that are used for the main system and the relevant independent subsystems, furthermore, the architecture of these subsystems will also be mentioned.

The strategies employed (along with relevant brief description are):

1. Use of Java as the programming language for the development of the android system.
   - The use of Java over Kotlin can be justified by the amount of support that is available for the former. The Java programming language still exists as the primary language for android development. Furthermore, our system relies heavily on object-oriented concepts and hence the strong object-oriented concepts in use by Java match the nature of our system.
2. Creation of an interactive but user-friendly interface
   - Since the user-experience is a determining factor for the usage of this application, an interactive interface is the end goal. Keeping in mind the literacy level of the target user, the application needs to ensure ease of use. Furthermore, the management of several plantations by a single user will also be only made possible by ensuring sufficient interactive ability.

3. Deployment of the CNN model on the internet as a web server.
    - The CNN model is deployed on the internet as a web server in a trained form. This is done to ensure that each system does not need to individually train the model in question.
4. Hosting the database online rather than maintaining it on the phone locally.
    - The database is hosted online because the web server and the software system make use of it to stay in sync with one another.
5. Use of threads and async tasks to await results from independent services.
    - This ensures that the main UI thread is not held up in tasks that await data from the internet or interact with it.
6. Management of storage space and limiting user image upload.
    - This is a key consideration for the application. The system can store only so many users captured pictures (after analysis). There is a cost associated with the storage and therefore a limit must be set.
7. Use of PostgresSQL as the default database structure
    - The project will make use of PostgresSQL as the default database software. The decision is made because of the relational requirements and the standardization of PostgresSQL as the market leader.

6.4.1 Deployment of the CNN model on the internet as a web server

This strategy warrants further explanation. A key question might be to ask why the application is not being shipped as a contiguous unit. The reason behind such relates to the nature of the application itself. The application must ensure performance to a reliable extent. CNN models are incredibly hard to train and take a lot of time as they figure out patterns on their own. If the models were maintained locally, this would induce two things: 1) no standardization of results, and 2) addition of a huge overhead cost related to processing. Therefore, the CNN is kept on the internet as a trained model. As a CNN is exponentially faster in prediction (compared to training), the trade-off in performance should be arbitrary.

6.4.2 Hosting the database online rather than maintaining it on the phone locally

The database serves the purpose of acting as a sync enabler between the android application and the web server. When a photo is captured by a user on the application, it is sent to two places: 1) the storage server, and 2) insertion of record in the central database. This record is used by the CNN on the web server to get the image path and download it. The image can be sent directly to the CNN server but since there is a need to upload it to a server anyway, this work around is used. Furthermore, the database server will contain the disease-mitigation relation records and the prime goal of this project is to ensure that these records are always kept updated.

6.4.3 Use of PostgresSQL as the default database structure

As the relationships in this project are of prime importance and consideration, therefore, a relational database was required to store the data. PostgresSQL was chosen as the software it comes with, Pgadmin4 is an easy to use, user-friendly database management system that can greatly help in curtailing and reducing time spent in database management. Furthermore, PGSQL has as of late become the market leader in relational databases. Due to this factor, ample technical support is available online.

6.5 Domain/Model Class Diagram

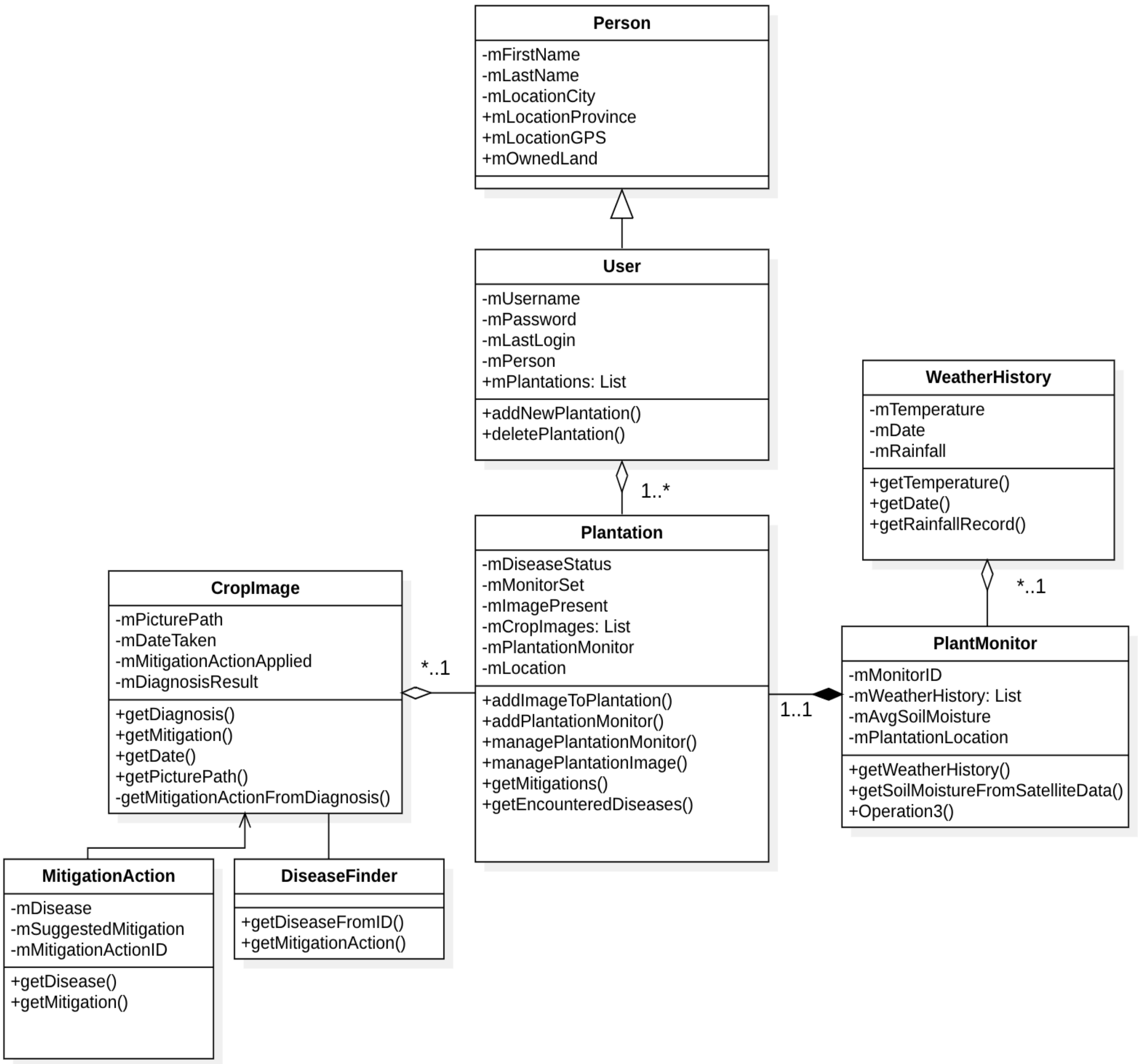

Figure 13: UML Diagram Representing the Software

This figure shows the class diagram of the logical classes used in the software.

6.6 Sequence Diagrams

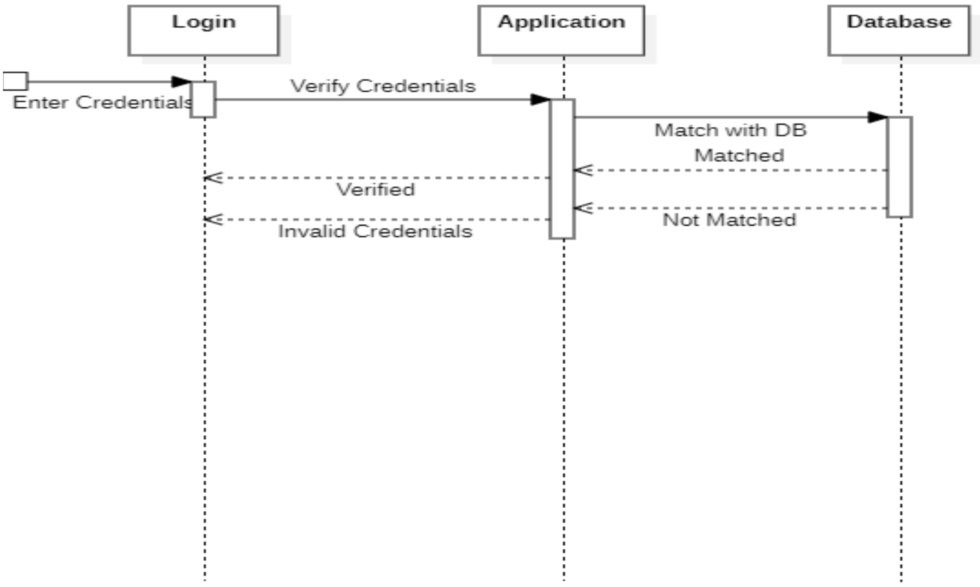

Figure 14: Sequence Diagram for Login Activity
This figure shows the sequence diagram for the login activity.

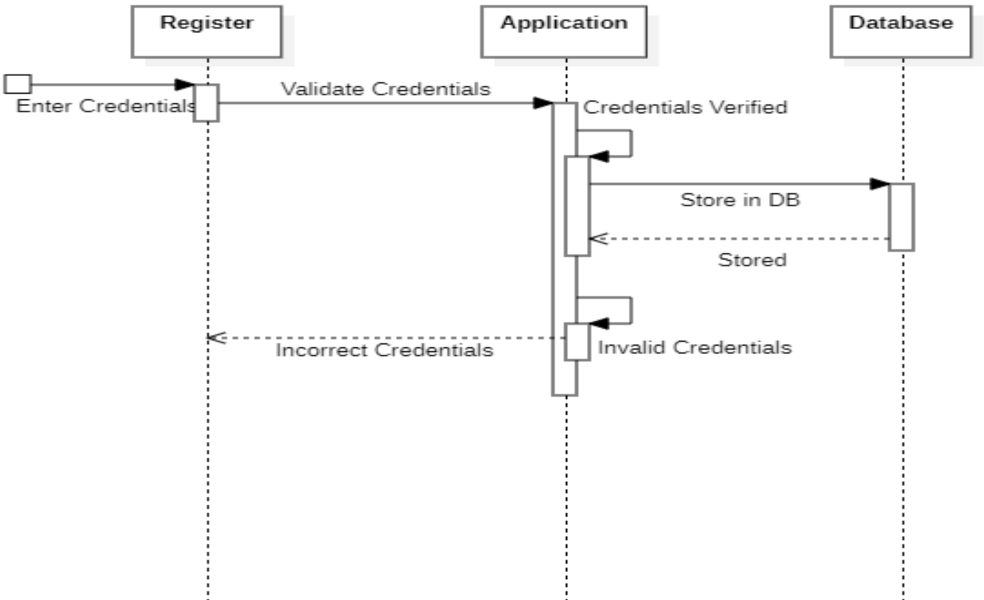

Figure 15: Sequence Diagram for the Registration Activity
This figure shows the sequence for the registration activity.

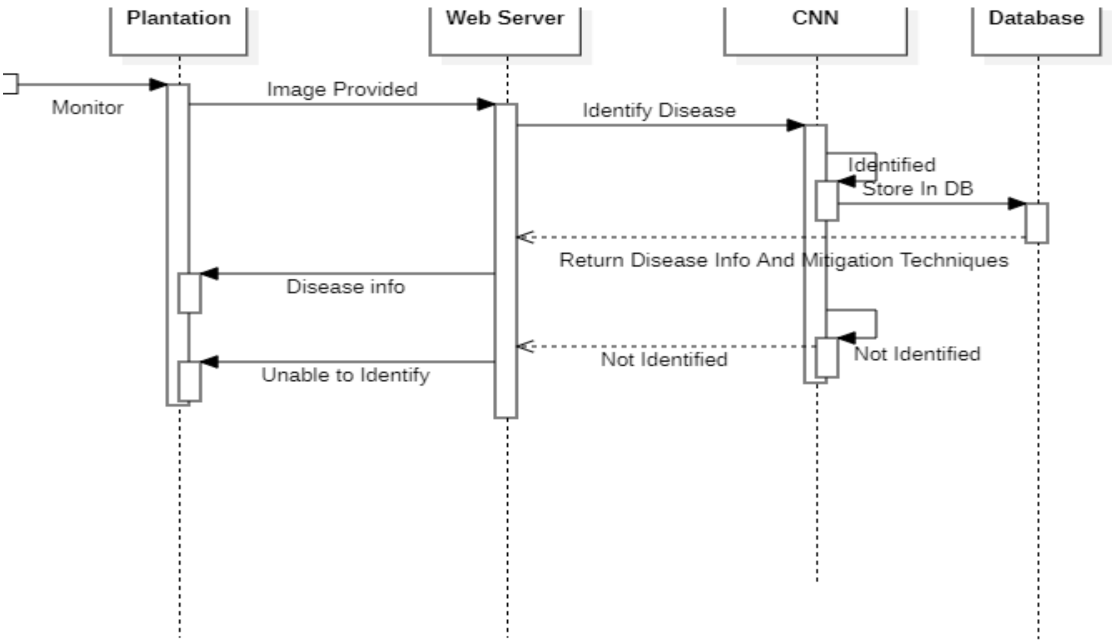

Figure 16: Sequence Diagram for the Detection Sequence
This figure shows the sequence diagram for the disease detection activity.

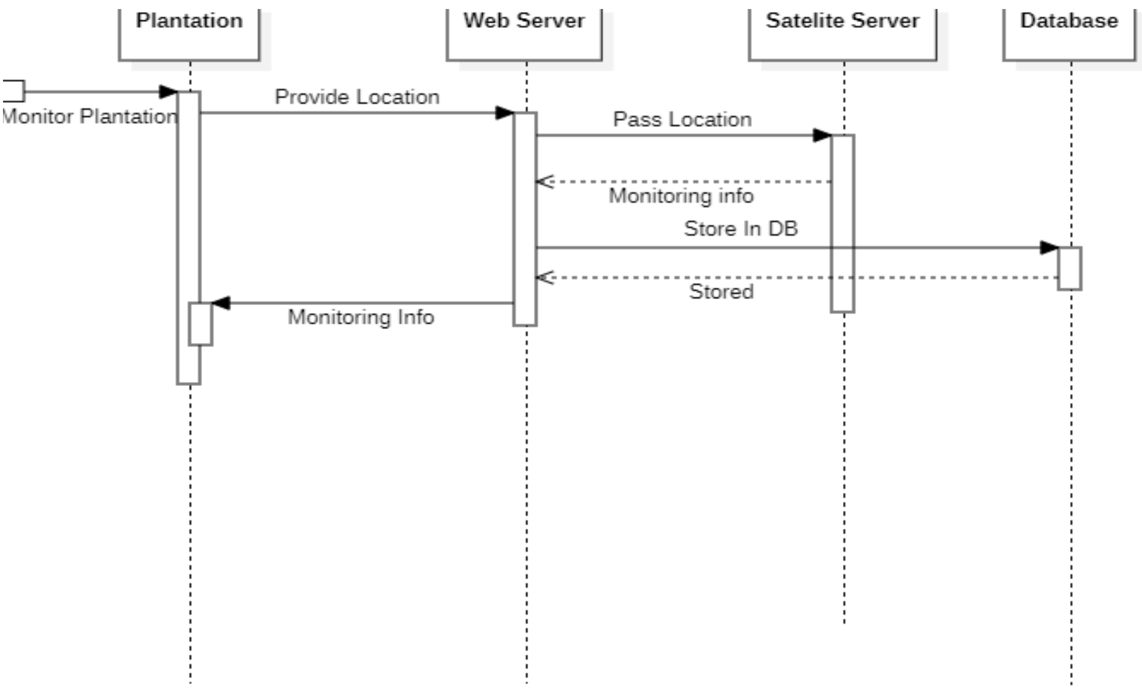

Figure 17: Sequence Diagram for the Monitoring Activity
This figure shows the sequence diagram for the monitoring activity.

6.7 Policies and Tactics

6.7.1 Use of Interfaces

The interface defines what can pass between the system and the environment. An interface is a description of the actions that an object can do. In terms of object-oriented programming, interfaces define the functions that an object can have. Unlike a class, an interface does not implement methods; instead, methods defined by the interface are implemented by the classes that implement the interface. A class can implement multiple interfaces. Interface allow us to take an overall overview of what the main functionalities will be related to. Use of interfaces enables abstraction and gives the flexibility to modify your program in the future. Interface makes unit testing easier**.** For instance, if your program is making use of interface for dependencies then you can easily replace those dependencies with a mock, stub, or a dummy for testing. Furthermore, interface allow faster development and increased flexibility. If you are coding as a team for a client-server application, use of interface will enable you to list down the main functionalities at first of each part which also improves the clarity and then implement them later on simultaneously without depending on the completion of one whole part while waiting for the other. Loosely coupled systems can be developed through interface. Polymorphic behavior can be implemented through them.

6.7.2 Maintainability of File Systems

To make sure that Hierarchical organization of the source code is maintained and that the code is up to date, it will be ensured that the file systems are maintained to fulfill this specific purpose.

6.7.3 Efficient Train-Test Split

The disease detection process will comprise of training and test the CNN, so it is able to identify the disease accurately. To ensure high efficacy we will be ensuring that the train test ratio is kept to 70:30.

6.7.4 Mitigation Testing

Another aspect followed by the disease detection will be identifying the relevant mitigation techniques. This will also be tested thoroughly to ensure that mitigation techniques are being stored and retrieved properly from the database against the specific id of their relevant disease.

6.7.5 Use of Java Coding Conventions

Coding conventions are meant to be followed by the programmers as it ensures quality, readability and maintainability of the code. Coding conventions of Java are mainly defined by Google and Oracle. Apart from google, Oracle’s coding conventions are considered essential as Java is owned by it. Some of the common Java coding conventions include:

- Name of the method should relate to the method's functionality.
- The variable name must be linked to its purpose.
- Declare global variables only if necessary to use in the other methods.
- Declare instance variable as private.
- Use getters and setters.

- Use getter and setter methods and avoid accessing variables directly from other classes instead
- All DB related code should be in the DAO classes only.
- The benefits of using coding conventions include low maintenance cost, improved readability and thirdly the combination of the previous benefits improves the overall pace of the software development process.

Chapter 7: Implementation and Test Cases

We have worked upon six components that are prototype will be composed of. These are Login, Register, Plantation Management, Disease Identification, Mitigation and lastly Monitoring. The functionality and details of each component will be discussed below.

7.1 Implementation

We have worked upon six components that are prototype will be composed of. These are Login, Register, Plantation Management, Disease Identification, Mitigation and lastly Monitoring. The functionality and details of each component will be discussed below.

7.1.1 Login Component

This is the primary component of our prototype. Here the user will enter their credentials to access the app. The text fields here used for the input will be Username and Password. Below these, there will be a button labelled as Login. Once this is clicked, the credentials will be compared with the usernames and passwords stored in the database. The output will depend upon whether the credentials are matched or not.

7.1.2 Register Component

Following component facilitates the new users to register themselves to access the prototype. This component will have three fields. These are Username, Password and Confirm Password. Once the user enters a username and matching passwords in both password fields, they will be compared with already stored usernames and passwords in the database. If they are not match, the credentials entered by the new user will be stored on the database and they can enter these credentials on the login page. “Registered successfully” will be the message returned upon successful registration otherwise an error message will be returned.

7.1.3 Plantation Management Component

A recycler view will be displayed where each recycler view item constitutes to each plantation of the user that has logged on to the app. Users have the option to click an item that is the plantation to further view its details, current diseases and their mitigation techniques. Users also have the option to Add or Remove the plantation.

7.1.4 Disease Identification Component

When the user accesses the plantation through the Plantation Management Component, they will have an option to select an image from gallery or take a picture to identify a potential disease that may exist in that plantation. Once the user clicks on the button of disease detection, the picture will be used to match the disease through our specified algorithm and the result will be displayed back to the user consisting of the details of the disease.

7.1.5 Mitigation Techniques Component

Mitigation techniques component will be displayed to the user once the Disease Identification component has been used. When the details of the identified disease are displayed, a button on the bottom right will be placed labelled as View Mitigation Techniques. This will redirect to the Mitigation Techniques component. This component will display the specific mitigation techniques of the identified disease. Only the mitigation techniques relevant to the identified disease will be displayed through this component.

7.1.6 Monitoring Component

The last component that our app will consist of is the monitoring of the plantation through satellite imagery for soil moisture and other useful information related to plantation. This component can also be accessed once a specific plantation is selected. When the plantation is selected, there will be an option or a button for Monitoring. Once this is selected the user will be offered to add location coordinates or enable location services to select the plantation area geographically. Once the user provides the plantation location through either of the options, this location will be passed onto the satellite parameters and the relevant results will be displayed back to the user through the Monitoring component.

Chapter 8: Conclusion and Future Work

So far, the work done during FYP-1 is based on the research part of the project. We have worked together to focus on multiple aspects of the project. We had to select the physical part of the plant that could majorly be used for the disease detection process. For that, we shortlisted down to the leaf of the plant due to its obvious dominance in portraying the symptoms of the majority of the cotton crop diseases. Furthermore, to shortlist the machine learning algorithm for image processing of leaf pictures for disease detection, we have selected CNN algorithm. This algorithm had numerous comparative advantages over the algorithms which is why it could not be overlooked. Adding to this, the monitoring of the plantation could either be done through the sensor-based networks or through the satellite imagery. Due to the high complexity of setting up sensor-based networks and maintaining their architectures that were difficult to be managed by the farmers, the use of satellite imagery proved to be a better choice. Combined with its high accuracy and low complexity in terms of use by the end user, this proved to be a more viable option for plantation monitoring. Furthermore, we have finalized the use cases, architecture, methodologies as well as the major architectural diagrams to be followed for development. The report covers a detailed analysis of requirements for the development of the project. The component breakdown of the system was portrayed through the presentation of high and level designs. The report further covered the detail of the prototype along with the details of each component that it comprised of.

As mentioned earlier, most of the research work has been completed under the scope FYP-1 time period. The future goals of FYP-1 remain regarding the pre-processing of the datasets and the final presentation of the prototype at the end of FYP-1. For the images that are not classified, we will use computer vision techniques to extract the features of the image that will eventually aid us in classifying those images. As far as FYP-2 is concerned, it will mostly be based on the development of the application through the architectural design proposed in FYP-1. The development will be done on android studio as our end users will be using the app on their mobile phones. Furthermore, datasets will be combined to produce larger datasets. Augmentation of datasets is also one of the main goals to be completed as part of data pre-processing in FYP-1. It is important to make datasets more efficient without the need of acquiring loads of data. The idea is that through the modification and changing of the makeup of data, one can improve the performance and enlarge the size of the training set. Moreover, the different modules that are discussed in the FYP-1 part will be integrated together in the application so that its previously mentioned functionalities are fulfilled. Finally, thorough testing of the application will be done to ensure that it is free of any issues and the desired outputs are obtained.

References

[1] Green Crop Ltd, "Cotton Diseases," Green Crop Ltd., 08 07 2018. [Online]. Available: https://greengroup.com.pk/cotton-diseases/. [Accessed 07 10 2022].

[2] Ayub Agricultural Research Institute, "Cotton Disease & Management," Ayub Agricultural Research Institute, 26 09 2020. [Online]. Available: https://aari.punjab.gov.pk/Cot-Dis. [Accessed 07 10 2022].

[3] Plant Village, "Cotton: Diseases and Pests, Description, Uses, Propogation," Pennsylvania State University, 31 03 2018. [Online]. Available: https://plantvillage.psu.edu/topics/cotton/infos. [Accessed 2022 10 07].

[4] Dekalb, Asgrow, Deltapine, "Identification of Cotton Diseases," Bayer Crop Science, 24 03 2021. [Online]. Available: https://www.dekalbasgrowdeltapine.com/en-us/agronomy/identification-cotton-diseases.html#accordion-432004-2. [Accessed 07 10 2022].

[5] V. A. Gulhane and A. A. Gurjar, "Detection of Diseases on Cotton Leaves and Its Possible Diagnosis," *International Journal of Image Processing (IJIP),* no. 5, pp. 590-598, 2011.

[6] V. Chaudhari and C. Y. Patil, "Disease Detection of Cotton Leaves Using Advanced Image Processing," *International Journal of Advanced Computer Research,* vol. 4, no. 2, p. 653, 2014.

[7] S. S. Patki and G. S. Sable, "Cotton Leaf Disease Detection & Classification Using Multi SVM," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 5, no. 10, pp. 165-168, 2016.

[8] C. U. Kumari, N. A. Vignesh, A. K. Panigraphy, L. Ramya and P. T., "Fungal Disease in Cotton Leaf Detection and Classification using Neural Networks and Support Vector Machine," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 8, no. 10, pp. 3664-3667, 2019.

[9] Caldeira, R. Faria, W. E. Santiago and B. Teruel, "Identification of Cotton Leaf Lesions Using Deep Learning Techniques," *Sensors,* vol. 21, no. 3169, pp. 1-14, 2021.

[10] S. Mohanty, H. DP and S. M, "Using deep learning for image-based plant disease detection," *Frontiers in plant science,* vol. 7, p. 1419, 2016.

[11] Brahimi, Mohammed, K. Boukhalfa and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence,* vol. 4, no. 31, pp. 299-315, 2017.

[12] P. S. Sreenivasan, "A study of cotton grown under constant soil moistures," *Proceedings/Indian Academy of Sciences,* vol. 30, no. 3, pp. 134-154, 1949.

[13] G. Naheed and G. Rasul, "Recent water requirement of cotton crop in Pakistan," *Pak: J Meteor,* vol. 8, pp. 75-84, 2010.

[14] J. Sehgal, "Soil-site suitability evaluation for cotton," *Agropedology ,* vol. 1, pp. 49-63, 1991.

[15] R. Kaur and O. Singh, "Response of growth stages of cotton varieties to moisture stress," *Indian Journal of Plant Physiology,* vol. 35, pp. 182-182, 1992.

[16] J. El Hachimi, A. El Harti, R. Lhissou, J. E. Ouzemou, M. Chakouri and A. Jellouli, "Combination of Sentinel-2 Satellite Images and Meteorological Data for Crop Water Requirements Estimation in Intensive Agriculture," *Agriculture,* vol. 12, no. 8, p. 1168, 2022.

[17] Z. Liqiang, Y. Shouyi, L. Leibo, Z. Zhen and W. Shaojun, "A crop monitoring system based on wireless sensor network," *Procedia Environmental Sciences,* vol. 11, pp. 558-565, 2011.