ABDULREHMAN AJMAL
01-134182-096
MAHAM KHAN
01-134182-022

Automated Weed Detection for Crop Health Monitoring

Bachelor of Science in Computer Science

Supervisor: Dr. Momina Moestesum

Department of Computer Science
Bahria University, Islamabad

July 2022

Certificate

We accept the work contained in the report titled "TITLE OF THE REPORT", written by Mr. AUTHOR1 NAME AND Mr. AUTHOR2 NAME as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Name of the Supervisor (Title)

Internal Examiner: Name of the Internal Examiner (Title)

External Examiner: Name of the External Examiner (Title)

Project Coordinator: Name of the Project Coordinator (Title)

Head of the Department: Name of the HOD (Title)

September 31st, 2015

Abstract

Smart agriculture provides an opportunity for a step-change in agricultural productivity. When it comes to crop yield, weed management is the most important aspect. They are the most impacting factors which act as a hindrance in overall crop productivity and causes important yield loss worldwide. Use of technology in the field of agriculture can help automate the process of weed detection. This will not only be highly efficient but also good for the environment. Our project provides one such solution in the form of Crop health monitoring system. In this project, images are processed and transferred to a deep learning model where the identification of weeds is performed using Yolov5. This model presents an alternative to the old traditional methods of weed detection and help the local communities improve their crop production and revenue.

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful. It is all due to His blessings that we got to complete this project. We would like to express our deepest gratitude to Ma'am Momina Moestesum who gave us the chance to work on this project. We are thankful for her supervision. Her guidance proved very helpful in making this project successful. She not only provided us with the reference material but also gave us very vital suggestions. We would also like to acknowledge Ma'am Maryam Bibi who taught us this subject and thank her for her knowledge which was crucial in its completion. We would also like to thank our friends Shahzaib Sajjad for his contribution and family who continuously motivated us to perform better. We offer our deepest gratitude to everyone who supported us along the way. May ALLAH bless them all (Ameen).

ABDULREHMAN AJMAL, MAHAM KHAN
Islamabad, Pakistan

July 2022

"We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn't anyone else."

Regina Dugan

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

GDP	Gross Domestic Product
SVM	Support Vector Machines
HOG	Histogram of Oriented Gradients
MLP	Multi-layer Perceptron
UAV	Unmanned Aerial Vehicle
PLS-DA	Partial least square-discriminant analysis
OBIA	Object based image analysis
DSM	Data Science Method
RGB	Red Green Blue
SLIC	Simple Linear Iterative Clustering
DCNN	Deep Convolutional Neural Network
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
VGG	Visual Geometry Group
YOLO	You Only Look Once
NDVI	Normalized Difference Vegetation Index
GUI	Graphical User Interface
LBP	Local Binary Pattern
API	Application programming interface
UI	User Interface
XML	Extensible Markup Language

Chapter 1

Introduction

Agriculture plays a significant role in overall GDP of any country[3]. Pakistan being an agricultural country is highly dependent on its crop yield. But unfortunately, factors such as weed infestation doesn't let the yield to be as much as it should be. Weeds are basically the unwanted plants that grow on the same space as crops. They not only occupy space which could otherwise be used for growing crops, but also affect the health of crops. Moreover, the population of Pakistan is increasing at a rapid rate. Pakistan is already ranked as 6th most populous country. An increase in population means an obvious increase in the need of food. Pakistan must increase its crop production. Agriculturists are continuously finding ways to increase the production and improve the yield. They are stressing on finding areas which have a room for improvement. Weed infestation is one of those areas. If weeds are managed, the production loss of crops can be saved meeting the food need as well as improving the GDP.

1.1 Background

The annual economic losses in the world caused by weeds infestation in the field of agriculture is estimated to be more than \$18.2 billion. \$12 billion of this amount is associated with the loss of production, \$3.6 billion is associated with the spendings on chemicals and \$2.6 billion is spent on other methods to control weeds[4]. The average crop loss due to weed in Pakistan is 11.5% which is greater than the loss of 9.5% in the world and estimated cost of potential crop losses is 300 million per annum. 265 weed species have been known to be blamed for causing economic loss worth 3 billion USD annually[5].

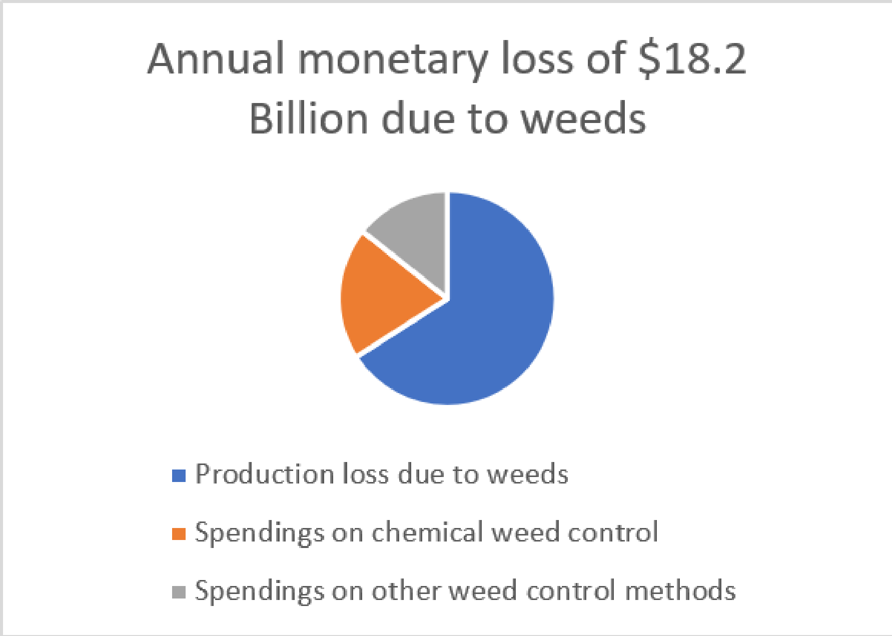

Figure 1.1: Division of Annual monetary loss

Agricultural experts have estimated the annual losses in grain yields of some of the main crops in Pakistan[4]. Their results which showed 18-25% loss in wheat, 20-65% in rice, 20-45% in maize etc Figure A.3.

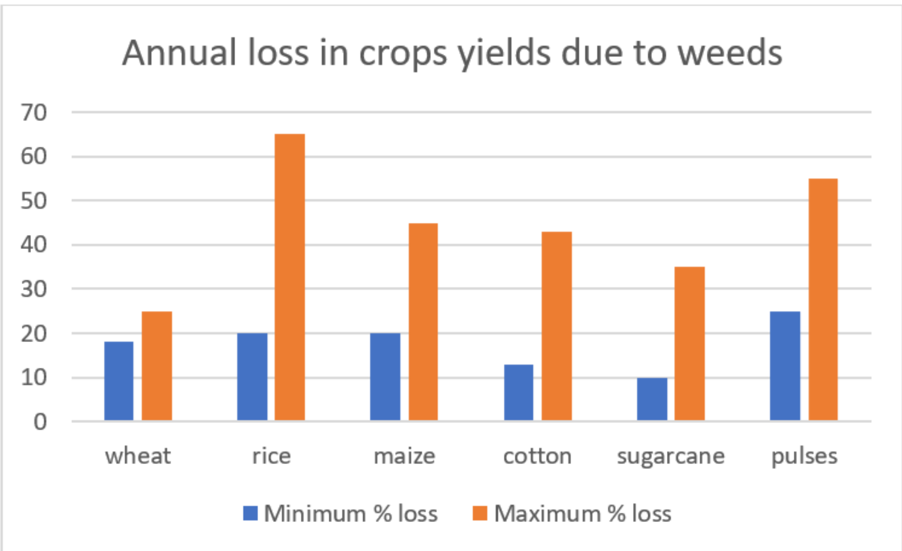

Figure 1.2: Average yield loss of crops due to weeds.

1.2 Problem Description

Weed presence reduces overall agricultural output by reducing total crop productivity as weed competes with other crops in terms of space, water, and nutrients.

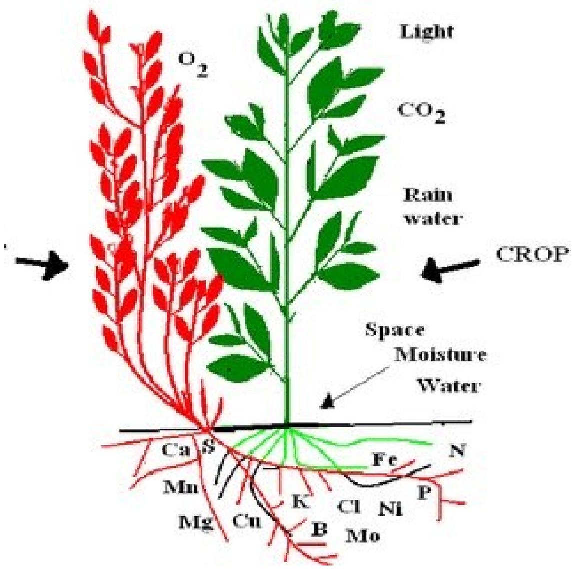

Figure 1.3: Weeds competing with crops in terms of nutrients[1].

Weeds also act as hosts for pests and cause many crop diseases. Apart from competing with crops, weed causes an increased risk of fire hazards. Moreover, they also make cultivation difficult because some of the weeds get entangled in machinery and act as a hurdle. Weed control is necessary to avoid all the problems caused by it. The three steps in weed management are:

- Prevention
- Eradication
- Control

Many different weed removal methods are being practiced from many years. They include cultural methods such as deep ploughing in summers and irrigation, physical methods such as the manual removal of weeds by hand pulling, cutting, and burning etc. moreover, chemical methods and biological methods are also practiced. Chemical methods include the spray of herbicides on the farm and the biological methods include the placement of natural enemies of weeds. Weeds are the greatest threat to agricultural production, causing a 45% loss compared to 25% loss due to diseases and 20% due to insects and pests with 10% by other factors [6]. Managing weeds has always been the biggest concern because it is quite an expensive and difficult task. Almost 1/3 of the total cost of crops production is taken over by weed management.

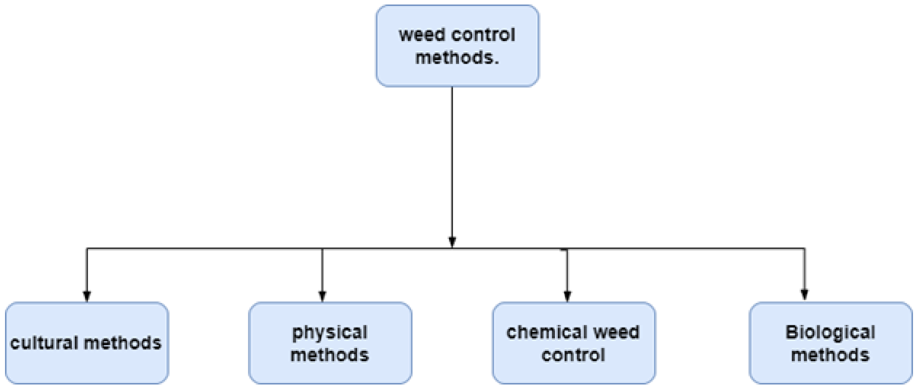

Figure 1.4: conventional weed control methods.

All these methods have some disadvantages. For example, the cultural methods do not control weeds they only provide a better environment for crop growth. These methods can lead to increased soil erosion. The disadvantage of the physical methods is that they are time consuming and labor intensive. Furthermore, by chemical methods the overall health of crops is affected, they cause pollution and are expensive. We are already at a high risk of Global warming and pollution index is also increasing day by day, so excessive use of chemical methods is dangerous. It is not only a waste of herbicides but also has health and environmental risks creating an imbalance in the ecosystem. Moreover, it also results in some species of weeds being resistant to it. There too much use is a danger to the health of all the living creatures. On the other hand, biological methods are very slow and less effective. These methods are outdated and very less effective, resulting in abundant problems in the field of agriculture.

We are in a dire need of smartly managing weeds. So, to prevent the risks and disadvantages of the conventional approaches mentioned above, we need an integrated approach that reduces the drawbacks. The use of precision technologies can serve as an asset in improving the agricultural yield.

An automated weed detection system can be employed. This system must be accurate enough to locate weed parts of the field. This use of computer vision in the agricultural sector has changed the traditional ways to manage crops. It has proven that resources can be saved at a large scale if utilized properly.

World is moving ahead in technology, and everything is moving towards automation. Weed control is one of those areas that demand automation. Previously, work has been done for using technology to solve agricultural problems. But there is still a gap that needs to be filled. The previous methods used to avoid this loss had disadvantages which can be improved by an automation in this field. An automated weed detection system for crop health monitoring using deep learning can come handy to improve the crop yield. The proposed project is to develop an application that uses pictures to detect weeds so that weed location in the fields can be identified as a result the whole farm can be saved from herbicides while saving time and labor work for farmers.

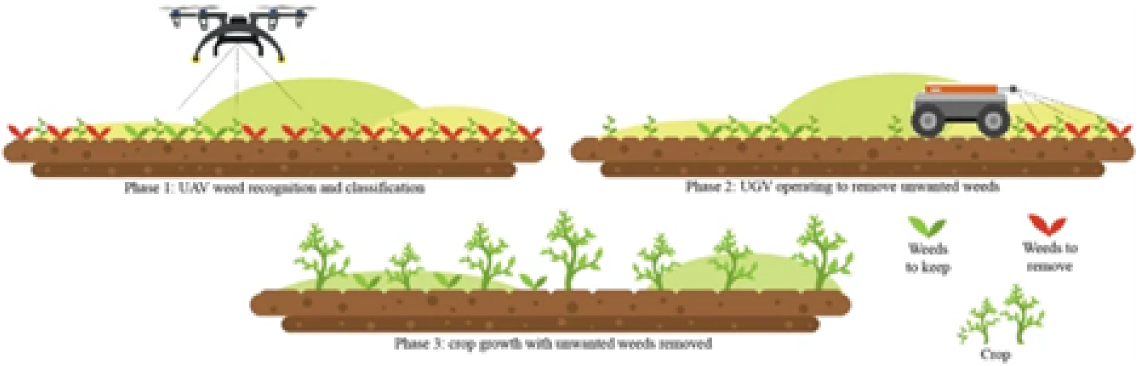

Figure 1.5: Overview of how automated weed detection may work.

1.3 Objectives

The objectives of the proposed project are:

- To detect weed and crops.
- To classify weeds based on its species.
- To find the ratio of weeds to crops.

1.4 Project Scope

The scope of this project is to detect weed from images that contain both crops and weeds, through deep learning model. Once the weed is identified it can be removed with the required equipment. Our project should be able to adapt to any plant types and be spatially accurate at sub-centimeter level, but its accuracy will be affected in dense areas where weeds and crops overlap each other.

Chapter 2

Literature Review

Automated weed detection is the need of the hour. The field on which our project is based on has been researched before. To improve the accuracy of weed detection, solutions have been proposed and many different methods have been observed. In this section we will discuss the contributions of articles in the field of automated weed detection and analyze them.

2.1 Traditional Computer Vision Techniques

In the past, many people researched on the use of machine learning algorithms to be used with image processing techniques to detect weeds. The traditional Machine learning methods can be used in agriculture for effective plant and weed identification. Use of machine learning in weed detection can prove to be a helpful approach as it require short training time and is less costly. A series of image processing procedures are used, and weeds are detected based on texture, shape, color, and spectral features. For example, Le et al. [7] discussed the difference in characteristics of corn and single species of weeds based on local binary pattern, texture features and SVM. Chen et al. [8] proposed a method built on multi-feature fusion and SVM for the detection of weeds. Experiments were conducted to study the effects of different features.

Moreover, Yuan el at. [9] proposed an approach which was based on combining the HOG features with SVM. In the proposed method HOG features of the leaf are taken and then the classifier is trained by SVM. It is then used to detect the leaves from the images.

In another article, a method was proposed for comparing the effect of SVM and Artificial neural network on detecting weeds [10]. This method was based on three shape features because sometimes it becomes impossible to be able to efficiently detect all the weeds from crops based on just a single feature.

Li et al. [11] projected a technique that used hyperspectral imaging data and machine learning. It was proposed to explore the fast and accurate detection of weeds in ryegrass and clovers. The images from two grasses and two weed species were collected. Preprocessing was done on them and then three classification models were used to train them. the three classification models were partial least squares-discriminant analysis (PLS-DA) , SVM and Multilayer perceptron (MLP) based on Sp model. All these showed fair amount of detected weeds but MLP based on Sp method showed highest accuracy.

In 2018, an article [12] on early weed mapping using UAV imagery was published. In this article, De Castro suggested an innovative approach OBIA, using RF algorithm to detect weed in early growing period of herbaceous crops. This RF-OBIA algorithm was used to create SSWM maps of early season herbaceous crops. This method basically linked UAV images, DSM's , Orthomosaics and created a random forest [RF] classifier exclusive of user involvement. All these things combined helped the OBIA algorithm to self-train the classifiers and select feature values and do the classification of weeds in both between crop rows and within them. First, OBIA distinguished all the plants in the image and accurately projected the plants height based on DSM. Next, a balanced training set was randomly selected by RF classifier. This process didn't require the exhaustive training process. This made RF-OBIA algorithm time efficient and more reliable compared to the conventional classifiers.

In another article the researchers proposed a different model for differentiating between crops and weeds by using SVM with blob analysis to assess its functioning in an automated weed control system [13]. Blob analysis is basically a machine vision technique built on the analysis of consistent image regions. The goal of this algorithm was to identify crops and weeds based on their features such as RGB, length and centroid. The methodology proposed in this article was that image processing techniques were utilized to detect weeds and then the processed images were to be given to robots. The processing of images also required binarization. Threshold values were also calculated and by this was image was separated from background. During this research the data was divided into two sets which included a total of 72 samples for training and for testing it contained 8 samples. The results showed variation with an accuracy laying from 50% to 95%. Images that had separated weeds and crops had maximum accuracy while the images which contained overlapping leaves didn't give good accuracy. The three techniques (binarization, SVM and blob analysis) used in this whole research were observed to be adaptive because two completely different images with different size, weeds and backgrounds were used for the test. However, the machine learning methods are easy to use and there has been a significant improvement in their algorithms and usage but there are some problems associated with some of the traditional methods discussed above. They are unsuitable for large-scale rapid detection or classification of images in a natural environment, cannot deal in complex field environment and are a little slower. To overcome these, deep learning methods were researched.

2.2 Deep Learning

In the process of weed detection, the techniques based on deep learning have obtained good results. Deep learning methods are based on the involvement of spatial and semantic feature differences. The accuracy of weed and crop identification has improved immensely with the use of deep learning. In order to get an overview of the deep learning methods used previously for weed detection ,deep learning based weed detection studies are summarized below:
Allessandro et al. [11] explains in detail regarding Weed detection in soybean crops using ConvNets. This research for the detection of weeds was divided into 5 different stages. The first stage was the capturing of images of soybean crops and for this purpose UAVs were used. Image acquisition was done with the help of visible light RGB camera, and the images were captured at an approximate altitude of 4m above ground level to ensure efficiency.

The next phase was of segmentation. In this research the algorithm of Superpixels was used for this phase and the segments were extracted. In this research simple linear iterative clustering (SLIC) superpixels algorithm was used for the segmentation purpose. This algorithm basically works on grouping pixels of same color together. SLIC algorithm creates pixel groups with the help of k-means algorithm. This k-means algorithm was useful because unlike traditional procedures it performs search in less space. Manual annotation was then done on these segments. By this way an image dataset of soil, Soybean and weeds was constructed. This dataset contained a total of 15336 segments out of which 3249 were of soil, 3520 grass, 7376 soybean and 1191 weeds.

The next stage included the feature extraction. Extractors of color, shape etc. were used. The fourth phase was of training the classifiers and in this case ConvNets were used to do the job. Comparison of ConvNets was done with other classifiers to deduce its performance. For the comparison test Adaboost- C4.5, Random Forests [RF] and Support vector machine [SVM] algorithms were used. Two different cases were used, one with balanced data and the other with unbalanced data. In the case of balanced data, the results showed that ConvNets were better with a case consisting of few classes when a robust training set is provided. Furthermore, in the unbalanced case all performance averages increased for all algorithms. But in this case too, ConvNets presented superior results to all other classifiers.

The fifth stage was based on segmentation and classification of images. The use of ConvNets accomplished great results. In this set 99.5% accuracy was achieved. All the other algorithms used for comparison also provided good results, but ConvNets had the benefit of their results not conditional on selection of good feature extractors.

Jialin et al. [14] investigated the use of DCNN for weed detection and its feasibility was showed . The experiment was divided in two stages, Image acquisition and training & testing. Three different datasets E-maculata, G. hederacea and T.officinale were used by the authors. DCNN was trained using images that had only one weed specie. Neural networks were trained using both single weed specie and multiple weed specie. Multiple specie neural networks AlexNet, GoogleNet and VGGNet were trained. In both cases AlexNet and VGGNet showed better results compared to GoogleNet. It was deduced that both image classification and object detection DCNN can be used in the machine vision but training of image classification DCNN does not require the drawing of bounding boxes so is time efficient.

In 2020, another article was published for deep weed detector for precision agriculture [15] which emphasized on the need of precision agriculture and explained how deep learning is becoming as the tactic for detecting weeds. Though, it also highlighted the fact that training dataset with only deep learning classifiers doesn't help much in precision agriculture. It further explained that it does not point out the weed as it doesn't separate the background. So, an alternative method of using ResNet-50 and YOLO v2 object detector was suggested. Basically, in this article an approach which used outcome of one of the layers from ResNet-50 as an input to YOLOv2 was proposed. The given method was useful for not only precisely detecting weeds but also for classifying the weeds on their types. In this paper the aim was to detect 4 different classes of weeds: bluegrass, chebopodium, circium and sedge. From the experimental results found it was justified that fused ResNet-YOLO network was the best approach for precision agriculture.

In [16] another article the author proposed 3 different methods for weed detection which helped them draw a comparison. The three methods used were HOG-SVM, CNN- YOLO and Mask-R-CNN. In this article the steps described were as follows:

1. Vegetation detection
2. Crop detection
3. Weed identification and quantification.

This study differentiated between one machine learning algorithm and two deep learning algorithms based on their efficiency and effectiveness. However, the dataset used for this experiment was created by the authors themselves. Pictures were captured from a commercial lettuce crop farm. This capturing was first done 60 days after seeding and then done twenty days after the manual weeding. A total of 100 images was selected 2219 lettuces with different levels of weed plants.

For the first method (HOG-SVM), HOG technique was used to get the features from each image. A training process was carried on 1400 images to deduce the weed and lettuce classes of object using masks designed through NDVI and the OTSU method. NDVI index mask removes extra objects from the image such as ground etc. SVM is used to identify the objects belonging to lettuce class and then are stored in a new mask known as crop mask. Crop mask and NDVI index mask are multiplied to make sure that only weed plants are left in the images.

In the second method (CNN- YOLO), bounding box co-ordinates were used to separate crops from the images. Moreover, A green filter was used to binarize picture which resulted in pixels without crops to be converted into black, while the others to be white. The entire bounding box was considered instead of using edge detection. This affected the weed calculation because the nearest weed to the crop got ignored.

In the third method (Mask R-CNN), a complete image was processed to do the approximation related to weeds. NDVI index was used to remove background, the image was then mixed with masks provided by neural network. The % of weeds was then calculated.

According to this experiment, the author deduced that YOLO model was speedy and didn't have a significant effect on the evaluation although the closest weeds were ignored. Whereas the R-CNN method was very precise in locating the edges. Since HOG-SVM requires less processing capacity its results can also be said good.

Selvi et al. [17], carried out experiment for seasame crops with different weed species. The proposed technique used RGB images and converted them into gray scale. Neural network techniques (CNN) was used for feature extraction. Classifiers were first trained than tested. The experiment shows 95% accuracy for classifying using CNN.

Gao et al. [18] presented an approach that develops deep CNN on Yolov3 architecture. Variation is appearance of plants and different growth stages act as a hindrance when detecting weeds, in this case, convolvulus sepium in sugar beet fields. Dataset was made by capturing images from two sugar beet fields. Synthetic images were generated based on this training dataset and a deep neural network based on Yolo was designed. It presented better results when compared to Yolov3 and yolo.

Jin et al. [19] suggested a method which involved CenterNet Model along with the image processing techniques. A trained CenterNet model was used to draw bounding boxes around vegetables. Anything that fell outside the bounding boxes was considered weed. The proposed method focused on detecting the vegetables only so that all other weeds no matter how many species could automatically be discriminated. Now to extract weeds from background, color index-based segmentation was used. This technique can reduce the complexity of weed detection.

Luiz et al. [15] proposed a real time weed detection system based on the Yolov5 in 2021. In this work, the focus was not only on identifying weeds but detecting the ones that are resistant to the commercial herbicides. A custom dataset was used which consisted of 5 weed species that were resistant to Glyphosate. The YOLOv5 architectures were successful, and their detection was not conditional on light and background conditions.

The above discussed articles review the work of researchers and compare machine learning and deep learning methods for the detection of weeds. Deep learning seems to be a favorable technique because of its accuracy. In conclusion, the needed information that will be required for this project has been learned from these researches.

2.3 Literature Review Table

SL No	**Title of paper**	**Author**	**Tools and Technology**	**Methods**
1	Effective plant discrimination based on the combination of local binary pattern.(2018)	Vi Nguyen et.al[7]	Xilinx Zynq ZC702 , VITA 2000 camera	LBP with SVM classifier

Table 2.1 continued from previous page

SL No	Title of paper	Author	Tools and Technology	Methods
2	Weed and Corn Seedling Detection in Field Based on Multi-Feature Fusion and Support Vector Machine. (2020)	Yajun Chen et.al [8]	MATLAB , Core i7 GeForce RTX 206	Multi feature fusion with SVM
3	Detection of wine grape leaves based on HOG.(2016)	MA Yuan et.al[9]	Not mentioned.	HOG-SVM
4	Evaluation of support vector machine and artificial neural networks in weed detection using shape feature. (2018)	Abdolabbas et.al[10]	Not mentioned.	SVM or Artificial Neural Network (ANN)
5	Identification of Weeds Based on Hyperspectral Imaging and Machine Learning. (2018)	Yanjie Li1 et.al[11]	Not mentioned.	PLS-DA SVM MLP
6	An Automatic Random Forest -OBIA Algorithm for Early Weed Mapping between and within Crop Rows Using UAV Imagery. (2018)	I.De Castro et.al[12]	Not mentioned.	RF-OBIA algorithm

Table 2.1 continued from previous page

SL No	Title of paper	Author	Tools and Technology	Methods
7	Weed Detection Using SVMs.(2018)	Murawwat et.al[13]	Not mentioned.	SVM with blob analysis
8	Weed detection in soybean crops using ConvNets. (2017)	Allessandro dos Santos Ferreira et.al[20]	Weka software version 3.6.6,	Research proposed the use of ConvNets also known as CNN.
9	Weed Detection in Perennial Ryegrass With Deep Learning Convolutional Neural Network. (2019)	Jialin Yu et.al[21]	Not mentioned.	DCNN was used.
10	Deep Weed Detector classifier Network for Precision Agriculture. (2020)	Abdulsalam et.al[22]	Not mentioned.	Layers from ResNet as an input to YOLOv2
11	A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multi spectral Images. (2020)	Kavir et.al[23]	Mavic Pro Camera GTX 1060	HOG -SVM, CNN-YOLO, Mask R-CNN

Table 2.1 continued from previous page

SL No	Title of paper	Author	Tools and Technology	Methods
12	Weed Detection in Agricultural fields using Deep Learning Process. (2021)	Thirumarai et.al[17]	High-Res camera, Python	RGB to gray scale, then CNN used.
13	Deep convoluti onal neural networks for image-based Convolvulus sepium detection in sugar beet fields.(2020)	Junfeng et.al[18]	Nikon D7200 pyTorch	Deep CNN based on Yolov3 architecture.
14	Weed identification using deep learning and image processing in vegetable plantation. (2021)	Xiaojun Jin et.al[19]	PyTorch , Python Open CV GeForce RTX 2080	CenterNet model was used.
15	Real-Time Weed Detection using Computer Vision and Deep Learning. (2021)	Luiz Carlos et.al[24]	Google Colab, Pytorch	YoloV5 was used.

Chapter 3

Requirement Specifications

3.1 Existing System

In this section, we will take a look at existing research done in this area. A lot of work has been done on precision agriculture but software applications are lagging behind. Local markets contains huge void in this area while international agricultural markets have bloomed using these methods. In existing systems sliding window object detection was used like RCNN and faster RCNNS which are not up to toady's standards but then came algorithm yolo (you only look once) which revolutionised the computer vision and became the standard of object detection.

3.2 Proposed System

The proposed system is to develop an application on the principal of deep learning. Dataset containing images of crops and weed. We are Pre-Processing the data with auto orientation stretching to 416 x 416. Then use object detection model for a custom data set. We chose a model which will achieves high accuracy and is able to achieve detection on higher frame rates . System will be web application which will communicate an object detection API. System will be able to provide data for crops and information about pesticides.

3.3 Intended Audience

The target audience for our project will be agricultural community. Our project will help them by bringing automation in this field. Our project idea is to help the local communities which are lagging behind due to the use of old traditional methods by bringing technology to improve their yield.

3.4 Requirement Specifications

3.4.1 Functional Requirements

Functional requirements are given below.

- System should be able to read images.
- System should be able to detect weeds and crops.
- System should be able to generate an output based on detection.
- System should be able to provide information about crops and weeds.
- System should be able to send output based report to third party.
- System should be able to show recommendations.
- System should be able to pass recommended actions to third party.

3.4.2 Non-Functional Requirements

Non-functional requirements are given below.

- **Availability:** After the deployment of the web application. It will be available for use.
- **Reliable:** System should be accurate and reliable in detecting weeds.
- **Efficiency:** System should be efficient in performing all the tasks.
- **Re-usability:** System components should be reuse able in other modules and applications.
- **Performance:** Response time for object detection should be minimal.

3.4.3 Software Requirements

This application will use following software.

- **React:** React.js is framework of JavaScript used to develop front end of the web applications by using different libraries.
- **Node:** Node.js is framework of JavaScript used to develop web applications.
- **Python:** Pytorch is a machine learning library which uses python and fastapi will be used to access object detection model over the web.
- **Dockers:** Docker is used to strip all the unnecessary parts of the application so the performance can be increased.

3.4.4 Hardware Requirements

To use this application hardware requirements are following

- **Internet Connection:** As this application is a web application user must have a stable internet connection.
- **Device:** Any device which can run browser which supports web applications.

3.5 Sequence Diagram

Sequence diagram illustrates the sequence of different processes in the project.

3.5.1 Open Application

In figure A.2 it shows the complete sequence of how user will open the application through browser and how web browser will interact will application.

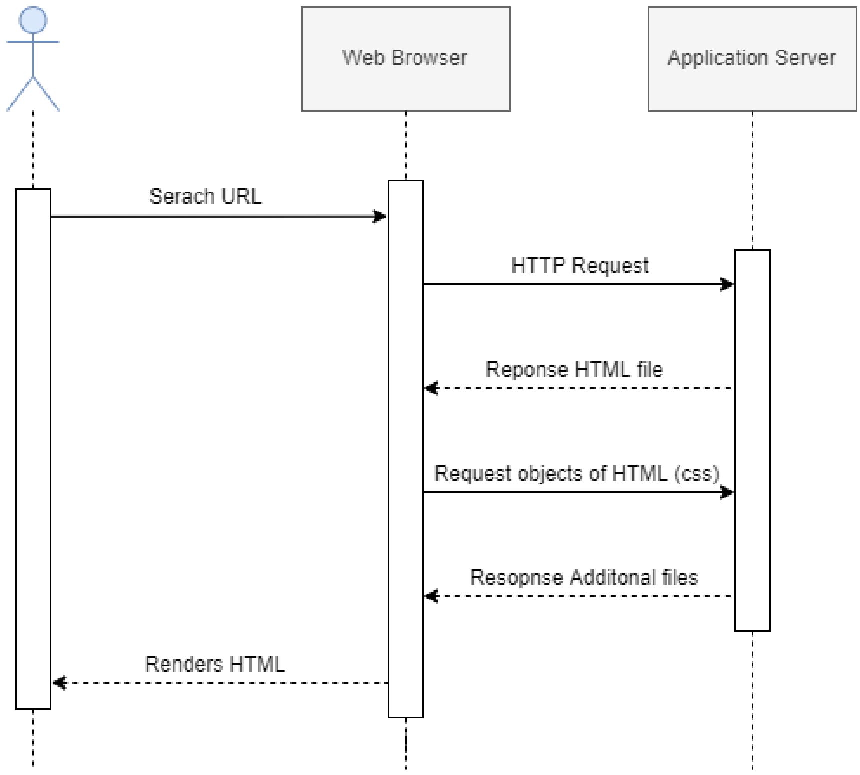

Figure 3.1: Sequence Diagram Open Application

3.5.2 Select Service

In figure A.3 It shows the complete sequence selecting service when the landing page will appear on the browser user will have to select which service or module user wants to work with this diagram show how user will interact with the application while selecting a service.

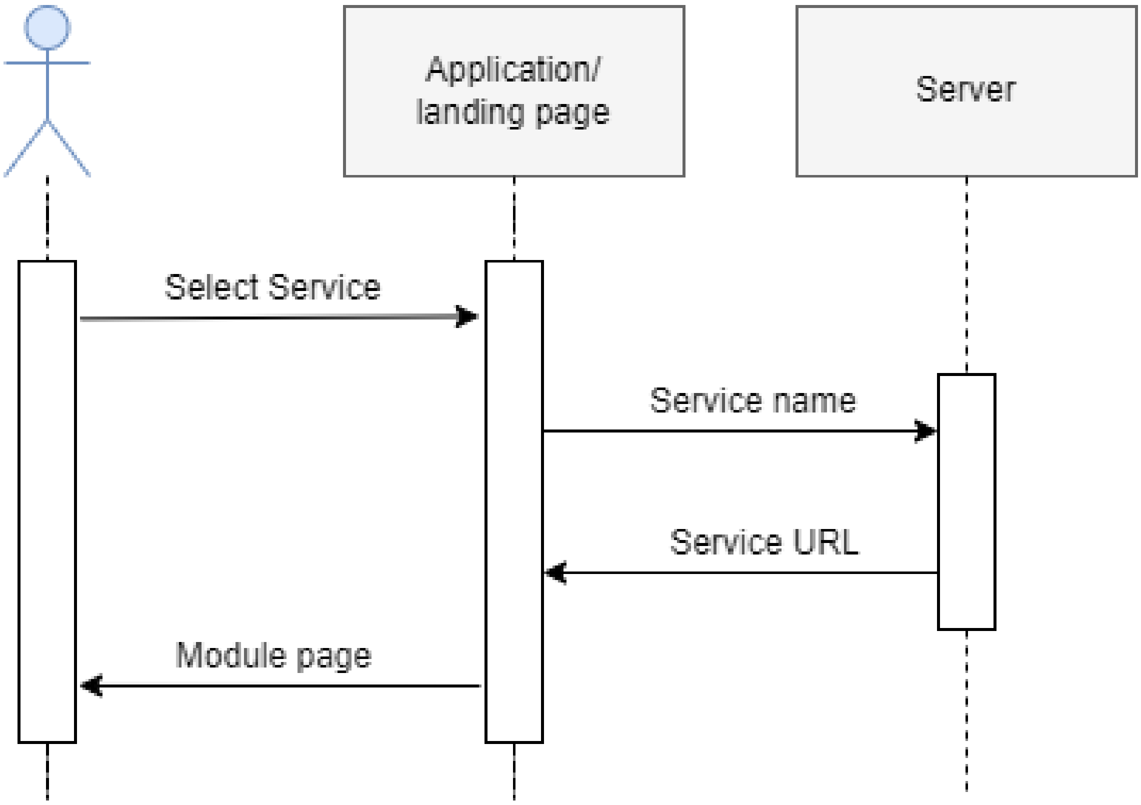

Figure 3.2: Sequence Diagram Select Service

3.5.3 Upload Image

In figure A.4 it shows the complete sequence of uploading image when user has selected a service system will not do any thing until user uploads an image this diagram shows how user will interact with the application while uploading the picture.

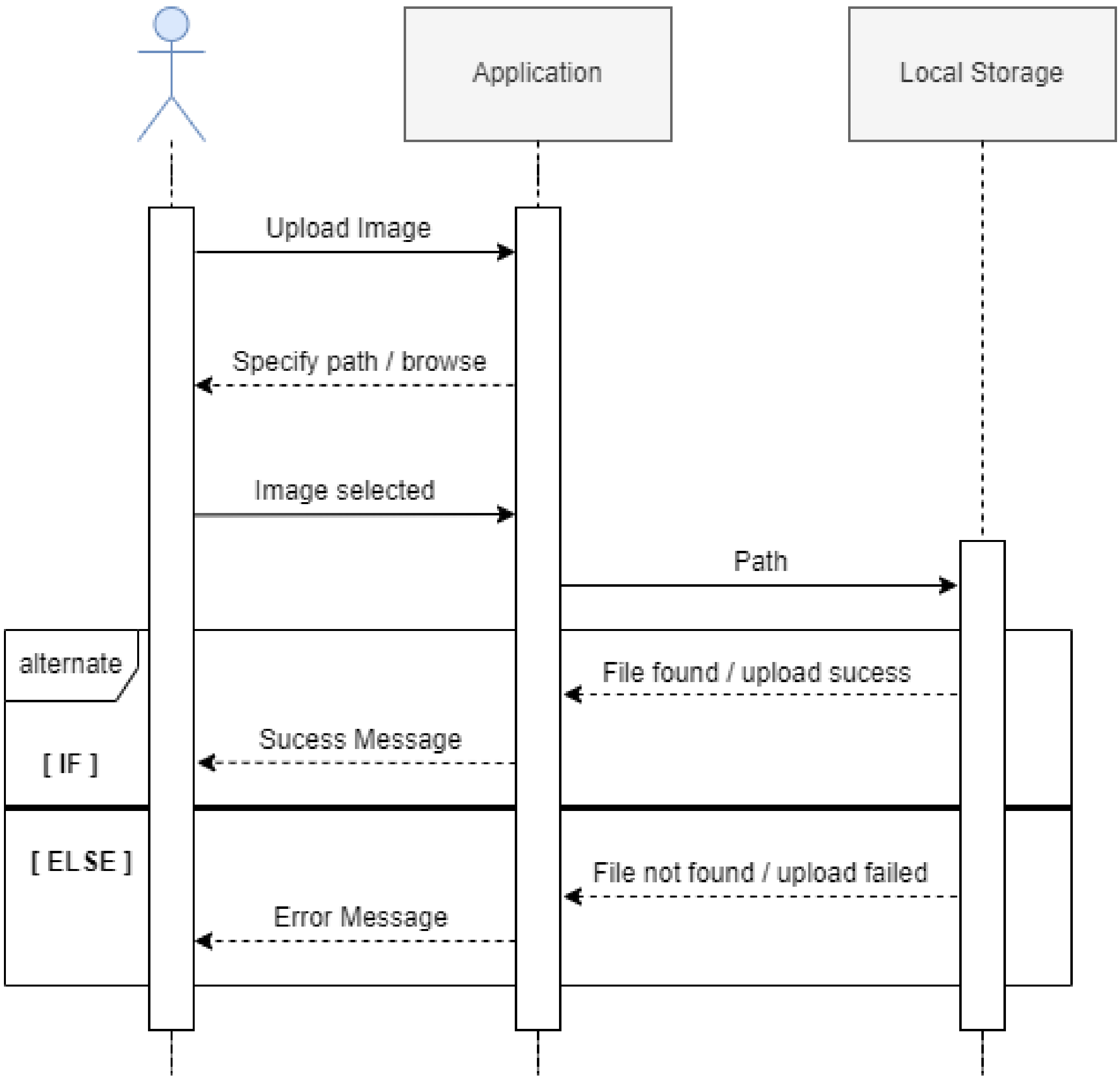

Figure 3.3: Sequence Diagram Upload Image

3.5.4 Weed Detection

In figure A.5 it shows the complete sequence of weed detection module this is our core module how the system will react when the image has been uploaded to the server. After successfully uploading how the image will intract with the core module of weed detection where with the help of object detection algorithm.

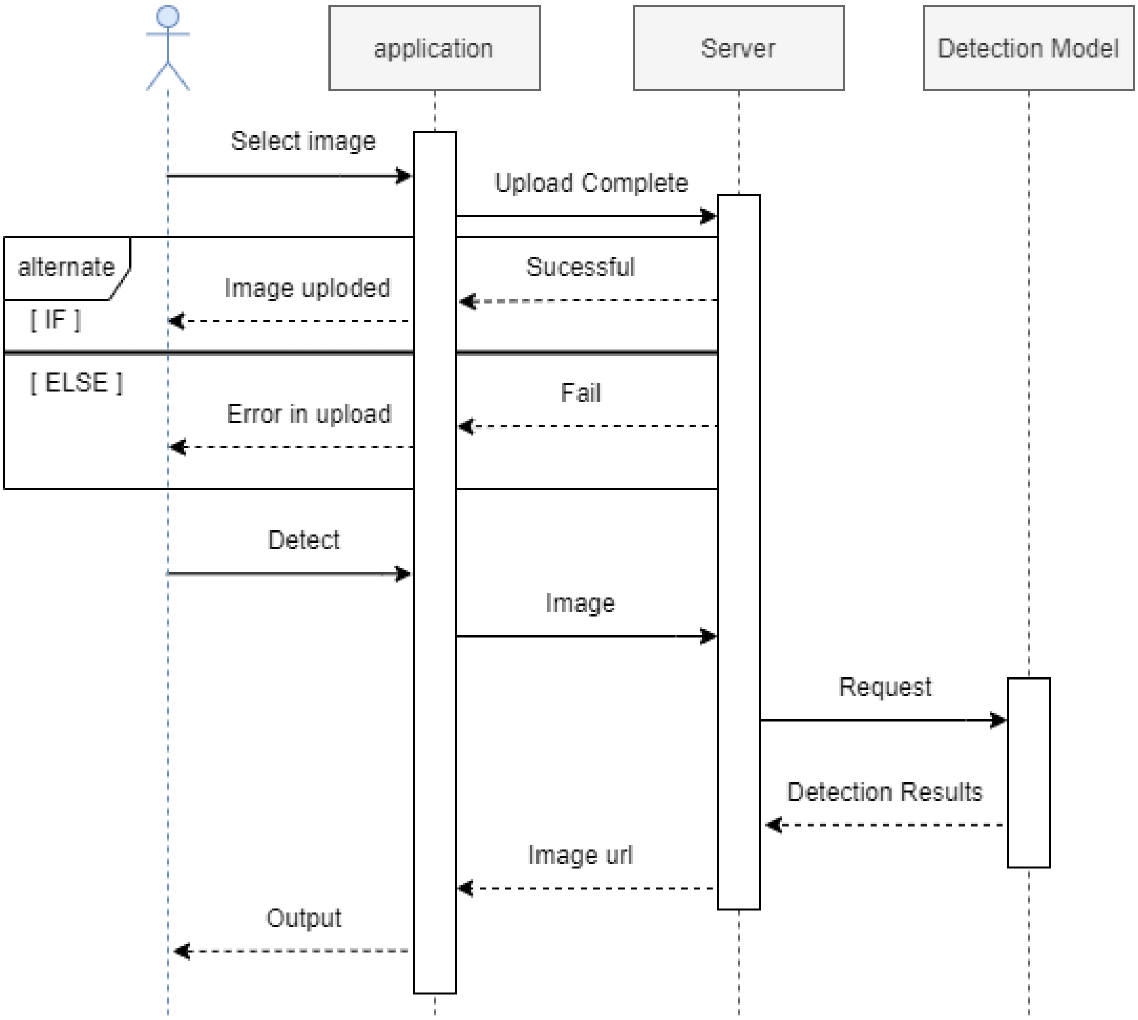

Figure 3.4: Sequence Diagram Weed Detection

3.5.5 Email Service

In figure A.6 it shows the complete sequence of passing the reports and recommendation to third party or lower staff.

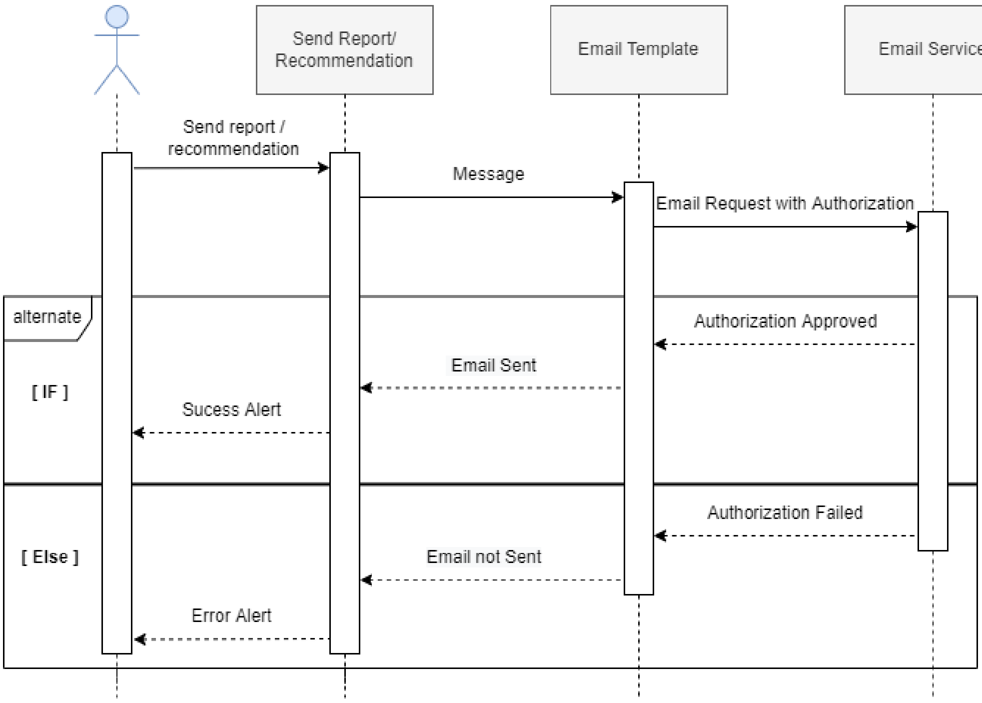

Figure 3.5: Sequence Diagram Email Service

3.5.6 Recommend List

In figure A.7 it shows the complete sequence of recommendation based on the service user selected and the detection results.

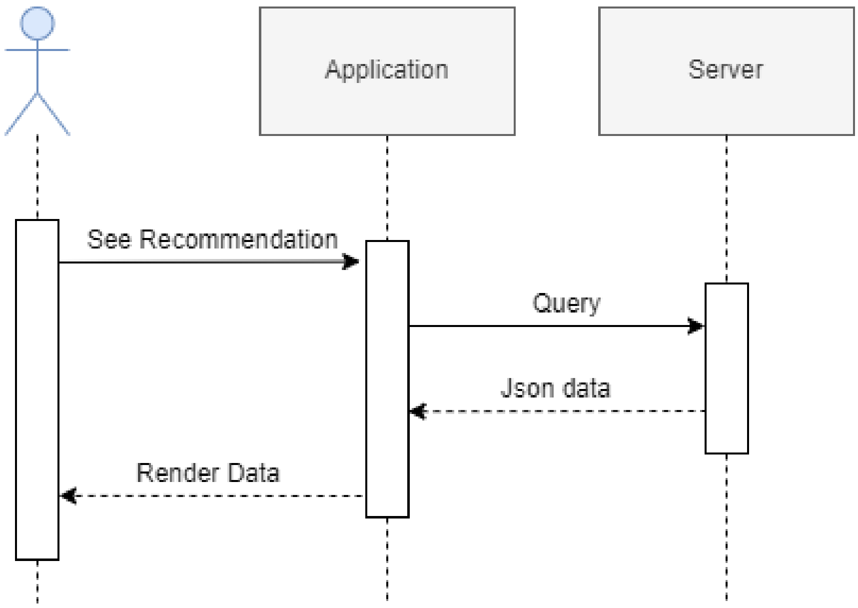

Figure 3.6: Sequence Diagram Recommend List

3.6 Activity Diagram

Activity diagram illustrates flow of activities from start to the end. This includes the flow of activity from user to the system and from system to components of the system.

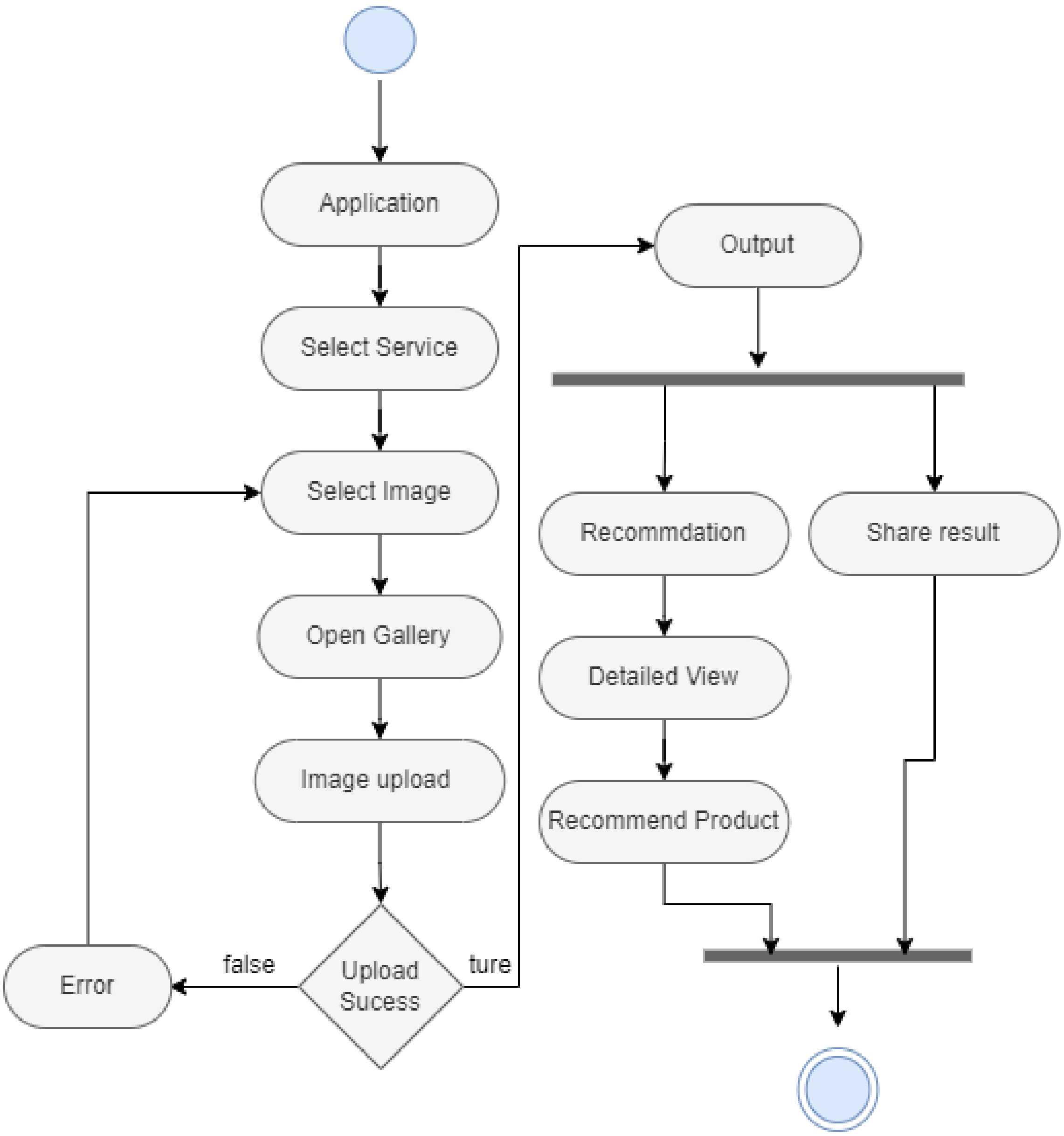

Figure 3.7: Activity Diagram

3.7 Use Cases

The goal of the user will be to monitor crops health by keeping a check on the weed count. User will be able to detect weed and crops. User will be able to gather information about the crops and weed. General use case diagram is shown in figure 4.8

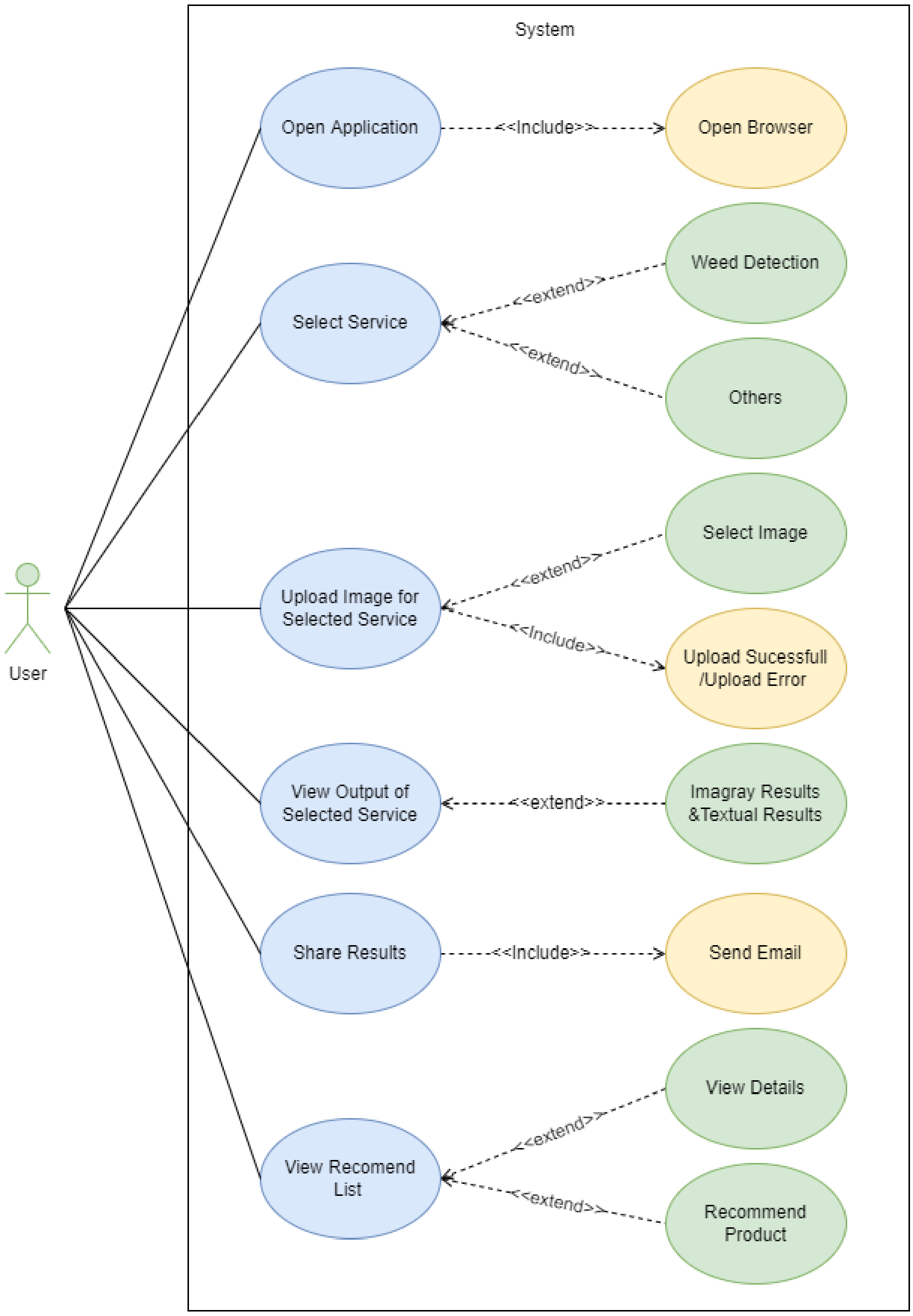

Figure 3.8: Use Case

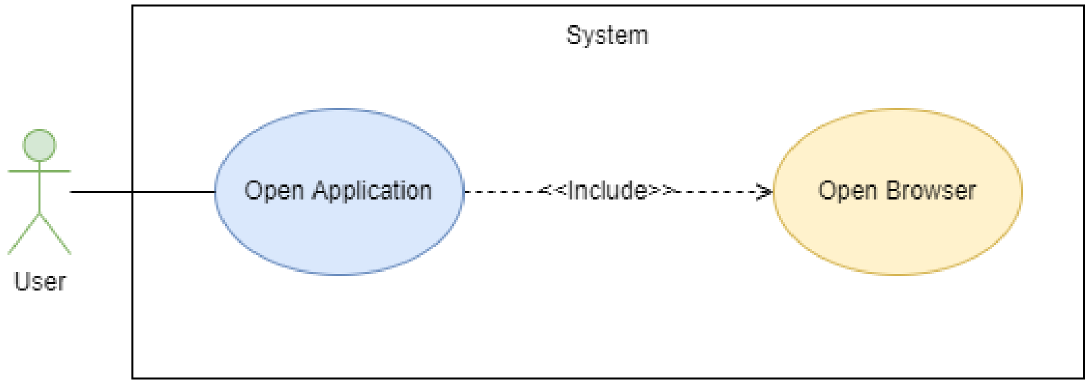

Figure 3.9: Use case open application

Open Application	
Title	Open Application.
Actor	User
Description	This use case represents the interaction between user and the web browser to start the application. User must have a working browser so that application can be run properly and smoothly.
Pre-Condition	Stable internet connection and web browser.
Post-Condition	Application landing page should display.

Table 3.1: Use case open application.

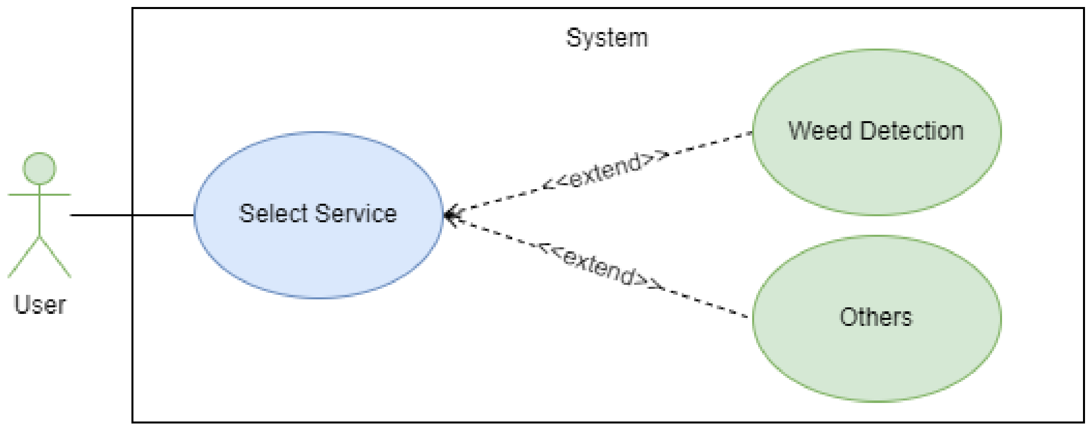

Figure 3.10: Use case select service

Select Service	
Title	Select Service.
Actor	User
Description	This use case represents the interaction between user and the application to select a service. landing page will have multiple options such as weed detection so user must select the a service in order to proceed.
Pre-Condition	Running application.
Post-Condition	New page according to the service.

Table 3.2: Use case select service.

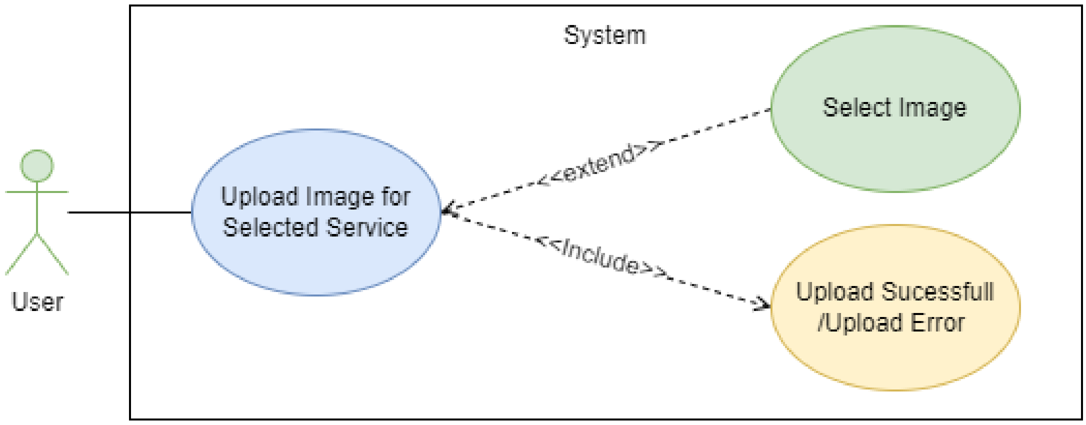

Figure 3.11: Use case upload image for selected service

Upload Image for Selected Service	
Title	Upload Image for Selected Service.
Actor	User
Description	This use case represents the interaction between user and the app to select the image file from the system. User must upload a picture in order to get the results. Upon successful or failed upload user will get a pop-up message.
Pre-Condition	Image file and internet connection.
Post-Condition	A successfully upload to application.

Table 3.3: Use Case upload image for selected service.

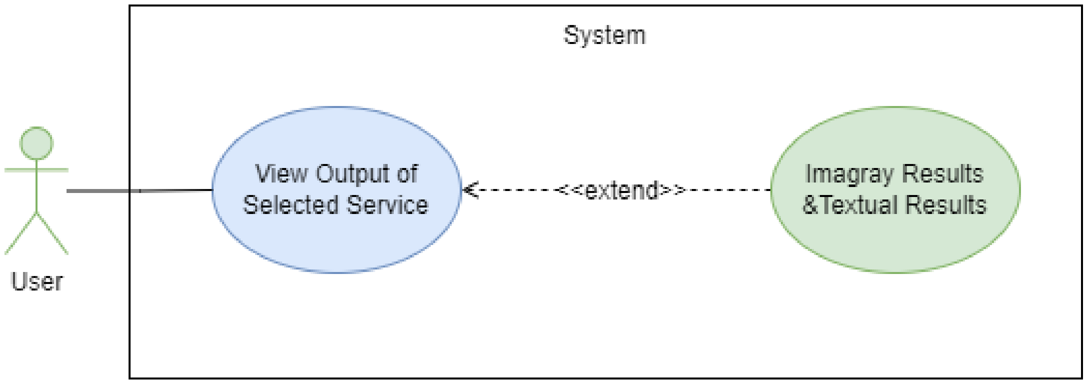

Figure 3.12: Use case view output of selected service

View Output of Selected Service	
Title	View output of selected service.
Actor	User
Description	This use case represents the interaction between user and the app to show output the image file after running it through detection model. Based on the service user will get to see results in images as well as some textual results.
Pre-Condition	Successful post request to backend.
Post-Condition	Output detection and bounding boxes.

Table 3.4: Use case view output of selected service

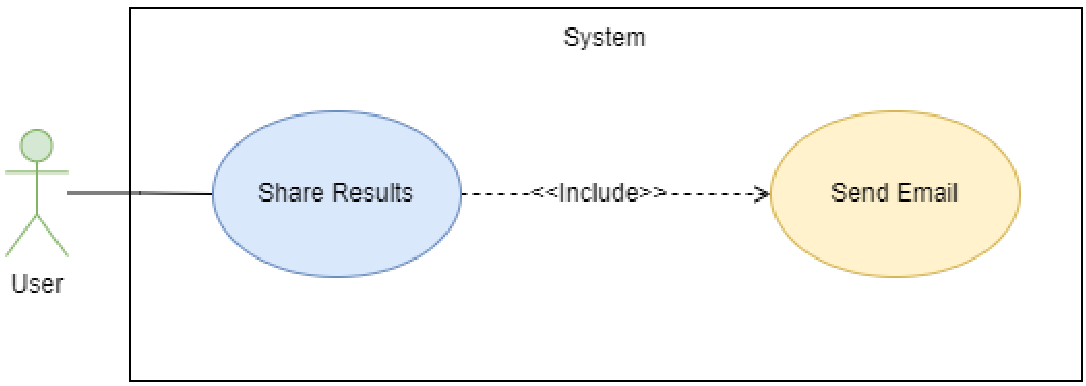

Figure 3.13: Use case share results

Share Results	
Title	View Report.
Actor	User
Description	This use case represents the interaction between user and the application to show report of crop health which has an option to send report to some third person.
Pre-Condition	Results from detection model.
Post-Condition	An email of crop report sent.

Table 3.5: Use case share results.

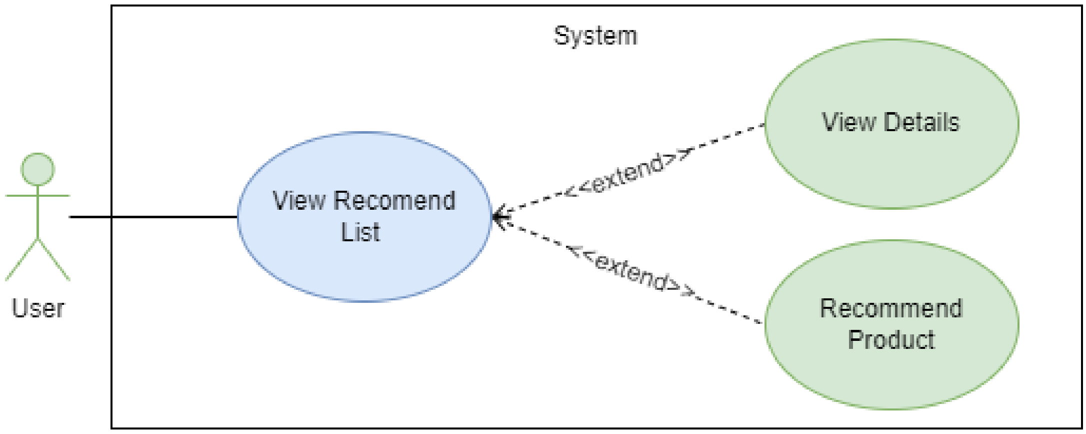

Figure 3.14: Use case view recommend list

View Recommend List	
Title	View recommend list .
Actor	User
Description	This use case represents the interaction between user and the application to Recommend some solutions with proper detailed description on the basses of detection results. Which has option to share recommendation to third person via email.
Pre-Condition	Detection Results.
Post-Condition	Output Recommendation list.

Table 3.6: Use case view recommend list.

Chapter 4

System Design

In this chapter, we have discussed the development phase of our weed detection and crop health system. The objective of this chapter is to get information about the basic flow of our application to convert the proposed design into a working solution. This chapter includes the proposed system architecture, used to describe the abstract view and conceptual model of our application. The design methodology and interface of our system are discussed in detail to get the idea of interaction between user and system interface. We have also provided information about our system components, modules, interfaces, and data to satisfy specified requirements in earlier sections. The chapter's detail discussed below:

4.1 System Architecture

The developed system is a web application which is based on 3-tier application architecture. User can only send input data to the application, all the processing is done on the server side for Weed detection. Each input image is divided into cells and each cell makes a bounding box and the confidence score then it predicts to which class it belongs. After processing data is transferred to database which is accessed and hosted by the server. The

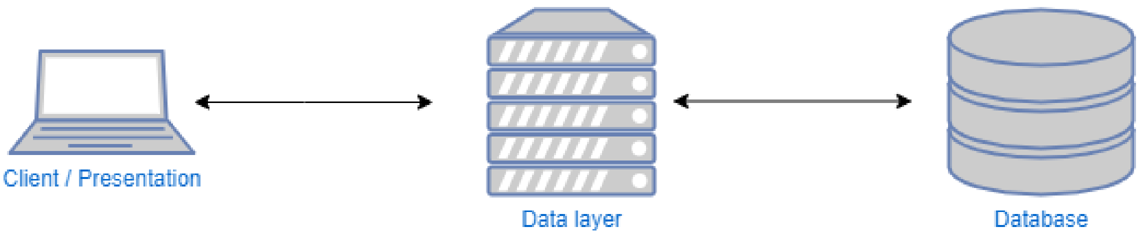

Figure 4.1: System Architecture

architectural view of the system is shown below. The illustration describes about the architectural design of the system.3.10

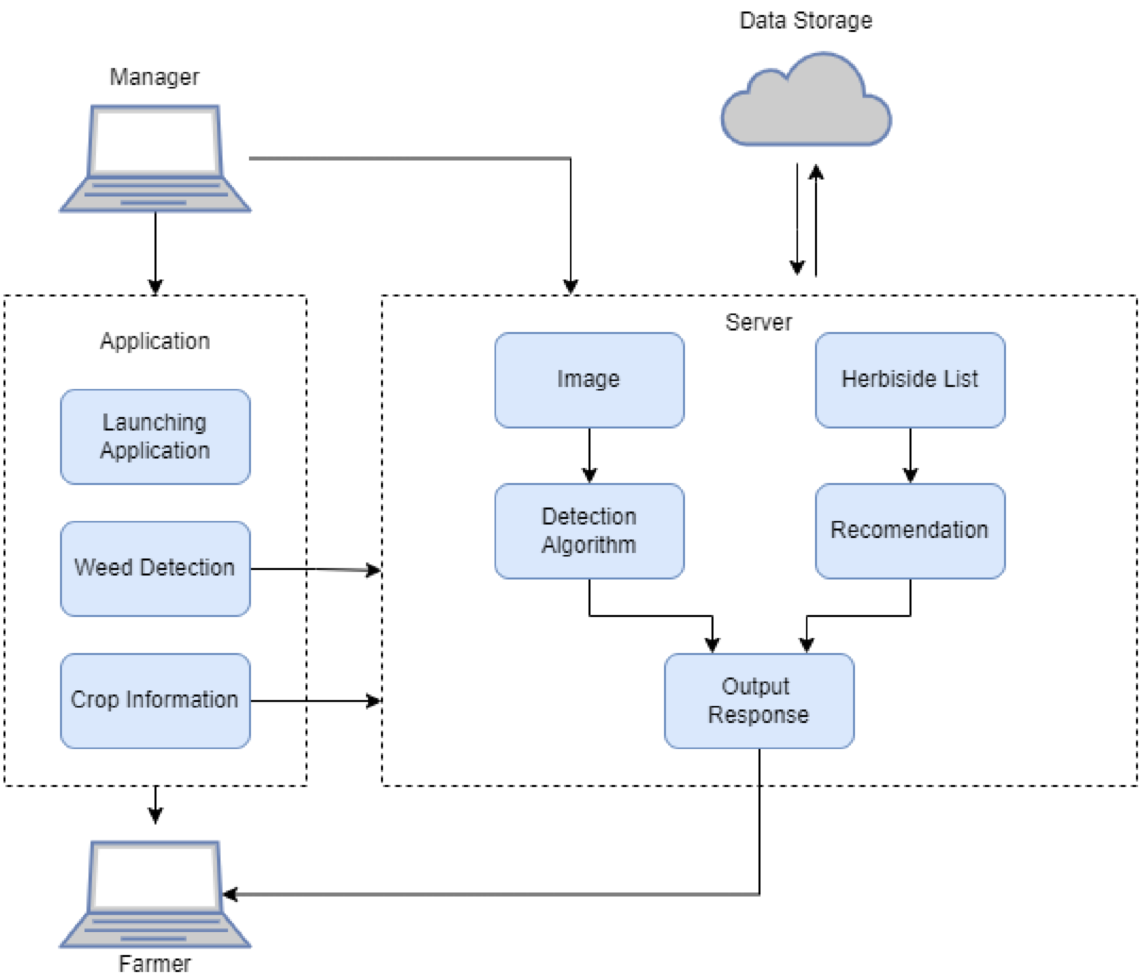

Figure 4.2: System Design

4.2 Design Constraints

The developed system depends highly on the quality of images if in some case weeds are densely populated in some areas or weeds and crops are overlapping, on each other systems efficiency can go down depending on the input image. By the quality of image it means that because the data is trained for land based data set solution it will not work for aerial data.

4.3 Design Methodology

The development of the project is directly linked to the software development model we are using. In this case, we are using iterative incremental model. Development of the project requires reassessment of the planning and implementation. After that we are designing the basic layout of the application and its features. After having a good idea of the requirements and specifications of the proposed project. Every iteration depends on the previous version of the iteration with each iteration there is a new module or better version of the product.

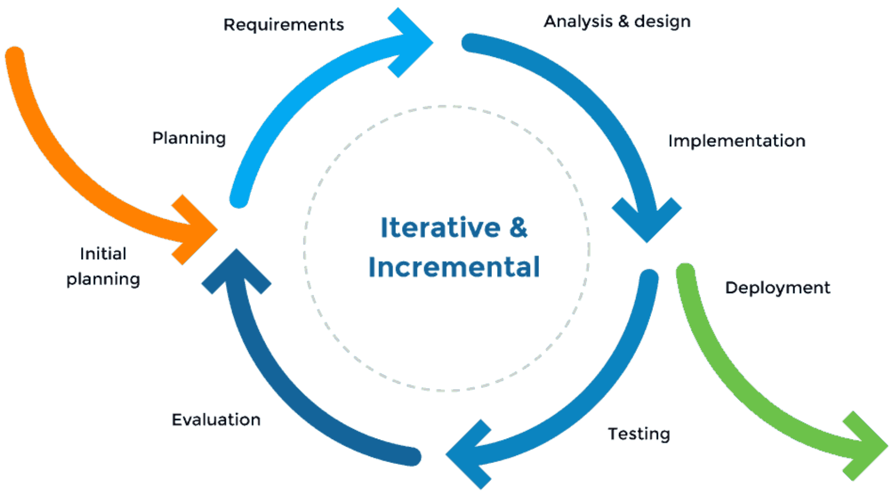

Figure 4.3: Iterative Incremental[2]

4.4 Context Diagram

Context diagram shows interaction of system and its actors. The illustration below shows the contextual view of the system. It provides boundaries between the system entities.

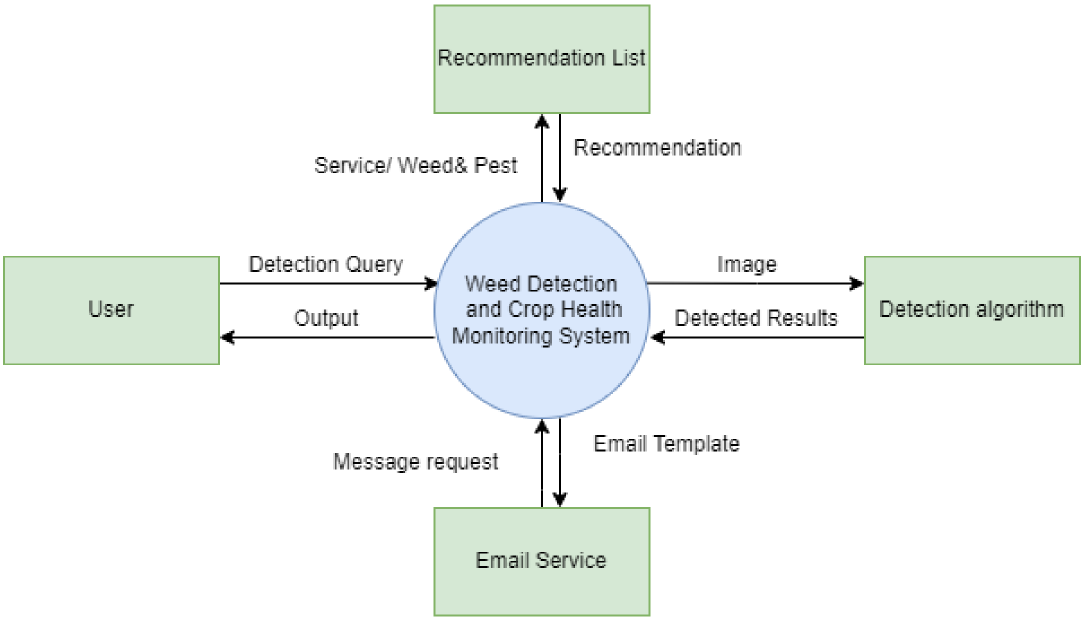

Figure 4.4: Contextual Diagram

4.5 GUI Design

System provides a easy to use graphical user interface over the web browser user can access the application from anywhere around the world with just internet connection. The application is Weed detection system and health monitoring system so GUI is built around in such a manner that the user can use the application without getting it self into the complications of technologies the user will have to upload an image and press a single button and the system will do its work. Interface of the application will be developed using JavaScript web frameworks.
Users should always be informed of system operations with easy to understand and highly visible status displayed on the screen within a reasonable amount of time. Interface designers should ensure that both the graphic elements and terminology are maintained across similar platforms. For example, an icon that represents one category or concept should not represent a different concept when used on a different screen.

- User will have to open web app through browser.
- User will then arrive at home screen.
- User will now have the access of various option regarding our application.
- Input is in the for of picture which can be accessed though the user's device.
- After pressing the detect button user will see an out put image of processed image.
- Under that user interface will provide stats of the image.

When the user presses detect button the client side sends a request to server side carrying that picture which will then processed by the detection model and in response server sends URL of the saved image.

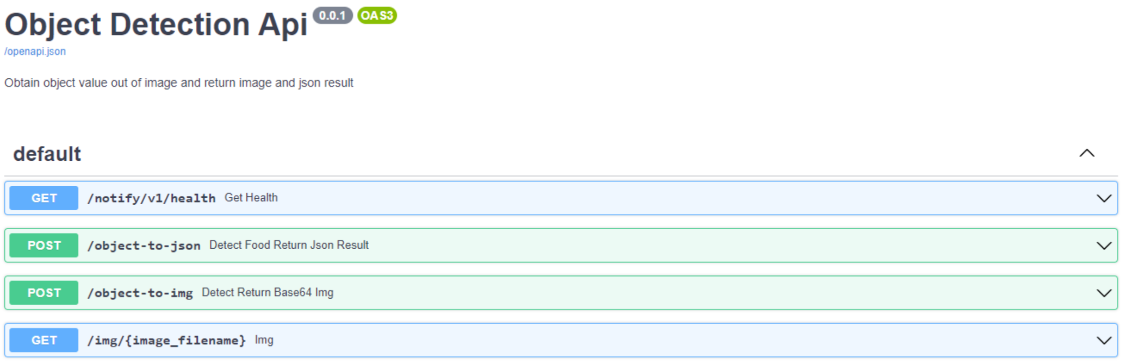

Figure 4.5: Back end Server

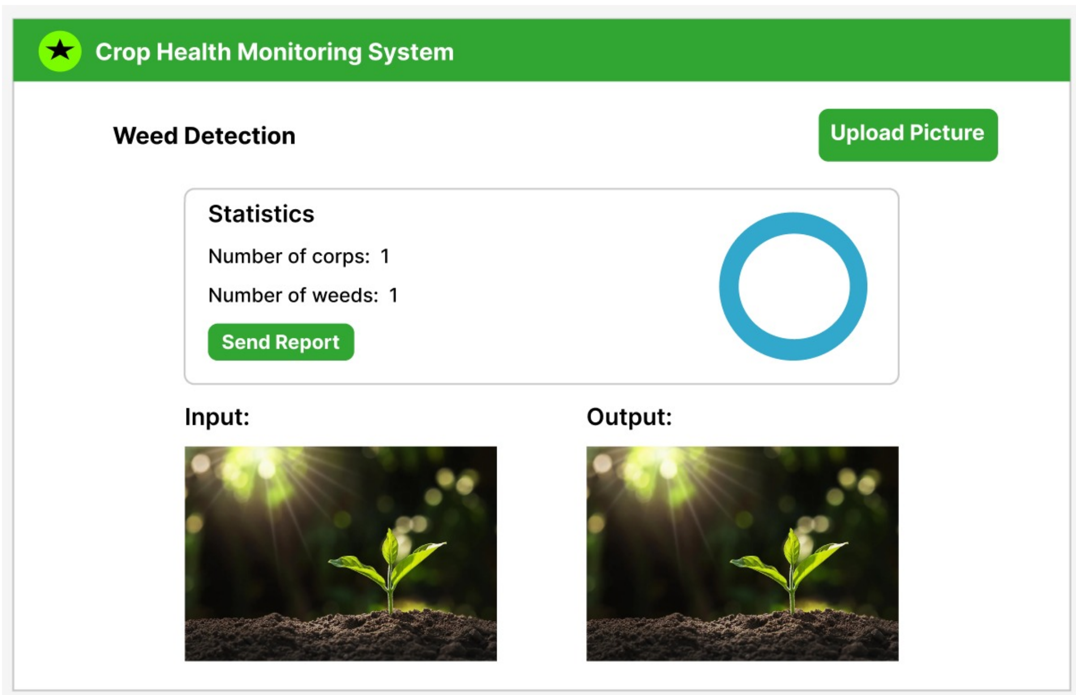

Figure 4.6: Detection page

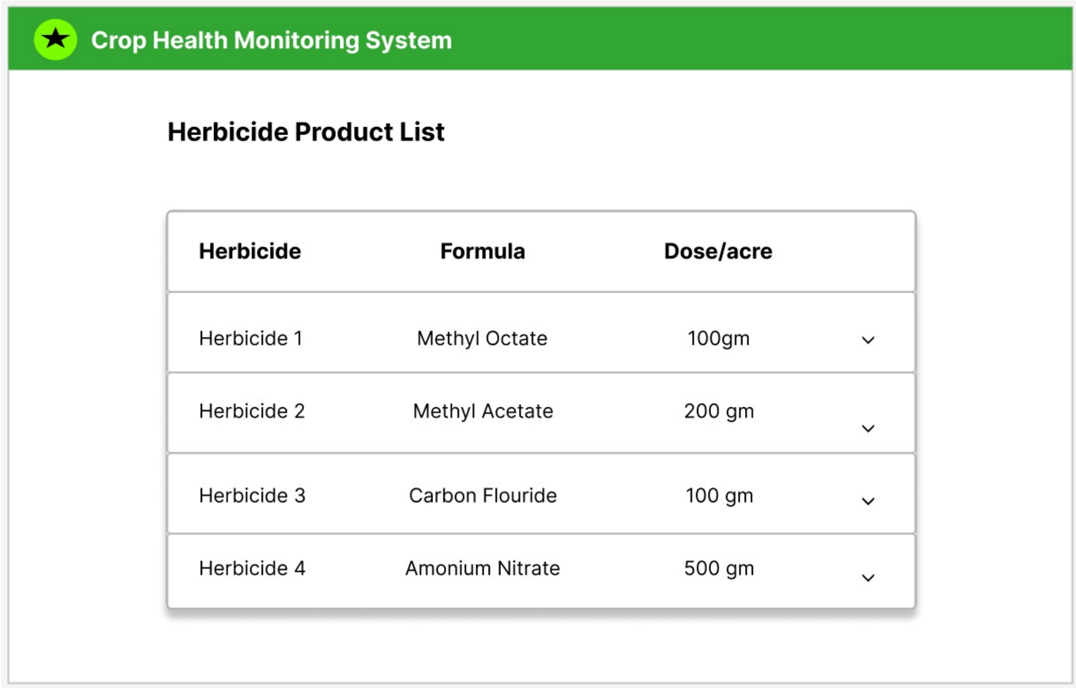

Figure 4.7: Recommendation page

4.6 Production Environment Design

In this section application's structure is explained how different technologies work together and how they are put together. In the following diagram. All the services run on the docker container because it has become a standard unit of deploying any server based application.

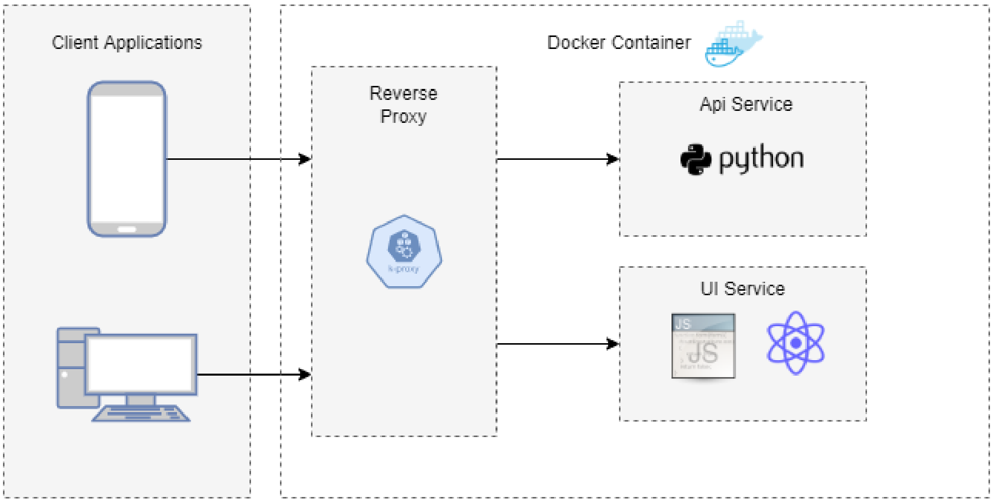

Figure 4.8: Production Environment

Chapter 5

System Implementation

System implementation includes all the stages involved in the deployment of our logical designs into a physical with the help of defined procedures and framework. In this phase, we develop system design to cope with requirements specified by the system as well as stakeholders and customers. In this chapter, we are going to discuss in detail all the tools, techniques, and procedures we have used for the implementation of prototypes, developed according to the system requirement in the system architecture phase.

5.1 Implementation Overview

Firstly we need a data set to train our model. We applied pre-processing techniques like auto orientation and we made all the images in the format of 416 x 416 and to increase the accuracy of the model we used the data augmentation techniques flipped vertically and horizontally.

To train our object detection model we used yolov5 object detection algorithm. Yolo is one of the most effective object detection algorithm which plays a crucial role in autonomous vehicle technology. Yolo works on CNN architecture which is extremely fast and achieves high efficiency. The architecture of yolo algorithm consists of three parts as shown in figure6.6. First is CSPDarknet which is a convolutional neural network used for feature extraction. Then it passes the output to PANET which is the second part which is used for feature fusion it is the combination of features from different layers so the proposed model can consume all the features. Then comes the final layer yolo layer which is also called the head shows the output of the detection result like classes and confidence score etc [25].

To pass our data to web we created a rest framework api so this weed detection api can be used in any application around the world. In application interface consumes the api so it can show results over the web.

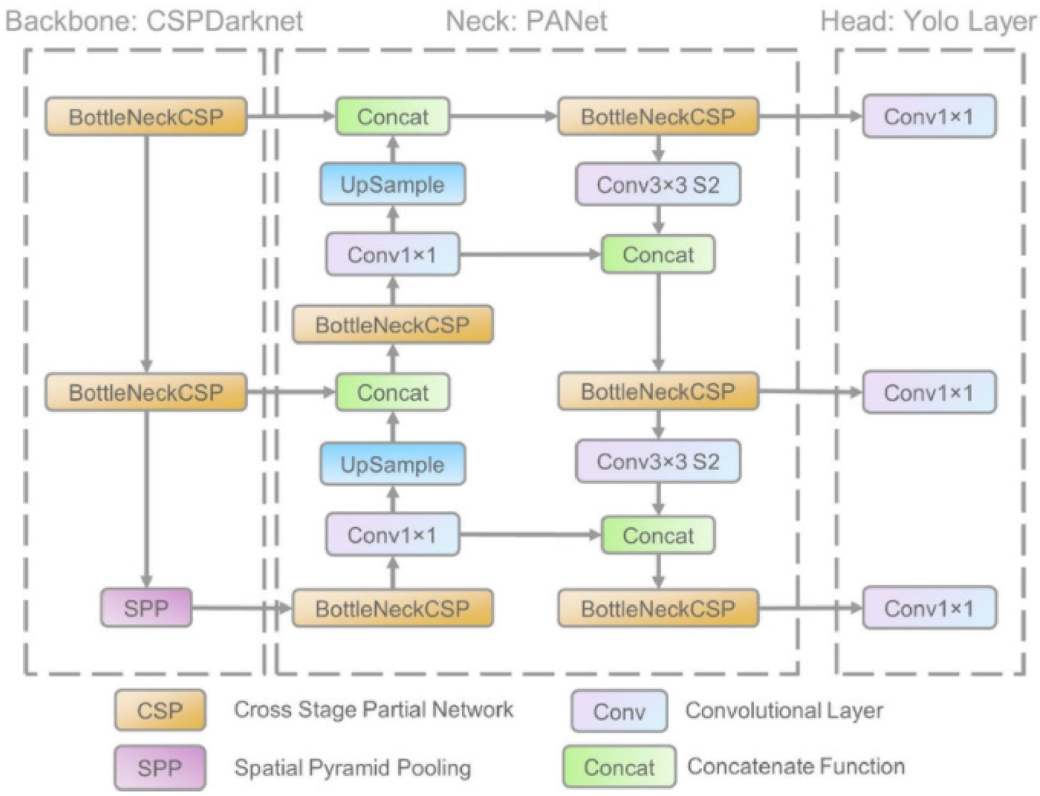

Figure 5.1: YOLO Network Architecture

5.1.1 Detailed Training Strategies of Model

We have trained a deep learning model using Yolov5. Two versions dataset were created one with augmentation and one without divided into 70% for training 20% for validation and 10% for testing. Annotations were converted from XMl to Yolo format. Images were resized into 416 x 416 stretched because it makes training much faster. We used google colabs which provided resources. Model was trained using NVIDIA tesla K80 & tesla T4.

Model	Train	Test	Optimizer	Batch Size	Epoch	mAP
Yolov5s	1900	118	SGD	8	500	0.702
Yolov5s	2500	118	SGD	16	500	0.790

Table 5.1: training strategies.

5.2 Data Description

Data set used in this project was of images of corps and weeds in their early stages of growth because system must be able to differentiate between crops and weed in early stages in order to control it. The data set used for this project is publicly available on mendeley data it consists of 1176 raw images in which 6 food corps and 8 weed types where identified and around 7853 annotations were there in XML format [26].

Abbreviation	Taxonomic classification	Family	Common name	Class
CA	*Chenopodium album* L.	*Amaranthaceae*	Goosefoot	Weed
GA	*Galium aparine*	*Rubiaceae*	Catchweed	Weed
TA	*Thlaspi arvense*	*Brassicaceae*	Field pennycress	Weed
CB	*Capsella bursa-pastoris*	*Brassicaceae*	Shepherd`s pursue	Weed
MI	*Matricaria perforata*	*Asteraceae*	Field chamomile	Weed
PC	*Polygonum convolvulus*	*Polygonaceae*	Wild buckwheat	Weed
VA	*Viola arvensis*	*Violaceae*	Field pansy	Weed
GP	*Galinsoga parviflora*	*Asteraceae*	Quickweed	Weed
BV	*Beta vulgaris*	*Amaranthaceae*	Common beet	Crop
DC	*Daucus carota* var. *sativus*	*Apiaceae*	Carrot	Crop
CPS	*Cucurbita pepo* subsp. *pepo*	*Cucurbitaceae*	Zucchini	Crop
CP	*Cucurbita pepo*	*Cucurbitaceae*	Pumpkin	Crop
RSS	*Raphanus sativus* var. *sativus*	*Brassicaceae*	Radish	Crop
RSN	*Raphanus sativus* var. *niger*	*Brassicaceae*	Black radish	Crop

Figure 5.2: Crops and weed types in dataset

5.3 Application Backend

This weed detection backend is developed with Fastapi this framework is the fastest python web framework. User sends server an image over the web using restapi which accepts and performs detection algorithm then returns image over the web. Package used with fastapi. This api consists of eight api each returns some information.

1. /notify/v1/health (Returns api's health).
2. /object-to-json-weed (Returns weed detection results in json format).
3. /object-to-img-weed (Returns weed detection results in image format).
4. /object-to-img-pest (Returns pest detection results in image format).
5. /object-to-json-pest (Returns pest detection results in json format).
6. /products/insecticides (Returns recommendation list of insecticides).
7. /products/herbicides (Returns recommendation list of herbicides).
8. /img/filename (Returns image from local storage).

5.4 Interface of The Application

For the implementation of Graphical user interface we have used react.js it is very popular open-source JavaScript framework for development of web applications. Packages used with react.

1. Material-Ui this is one of the best UI libraries for react.
2. Axios package is used to make api calls efficiently.
3. React-Router this is used to setup virtual URL links.

5.5 Additional Functionality

Our project has the capacity to incorporate new modules as required. For instance, we added an additional functionality of Pest detection and pesticide recommendation system in our crop health monitoring system. Pest infestation has been a problem for our crops from a long time. There is no perfect system to avoid pest infestation or save crops from the losses. Our system may help detect pests and may help in precision agriculture by locating pests efficiently and effectively and highlighting the areas which need immediate action. This way only necessary amounts of insecticides and pesticides will be consumed. This will not only save cost, time and labor but also help maintain healthy crops.

5.5.1 Working of Pest Detection

Dataset used during the implementation of pest detection module was not publicly available. It consists of 1300 images. Preprocessing was done the same way as for weed detection. Images were employed on YOLOv5 for training and object detection. YOLOv5 was successful in detecting pests from the images used. It had almost similar implementation as weed detection. This shows that our project is completely generic and can be very useful.

5.6 Tools and Technology

5.6.1 ReactJs

React makes it very easy to create web applications. it is very efficient and most commonly used library of JavaScript. It is very easy to build encapsulated components which can be rendered and re-rendered efficiently using state and stateless components. It can help in creating new functions without having to rewrite code again.

5.6.2 Recharts

Recharts is react library which makes creating different types of graphs very easy and unlike other graphical libraries which recharts is very lightweight as compared to other libraries.

5.6.3 Fastapi

FastAPI is a modern, high performance framework for developing API's it is the fastest python framework for web applications. It reduces development time by 100% to 200%.

5.6.4 EmailJs

Emailjs makes it very easy to send emails from JavaScript which otherwise require a very long process of sending and receiving emails and integrating email templates for every emailing service. This framework works very well with react and makes it very simple to create email templates.

5.6.5 Docker

Docker is a container that provides us the facility to run our application in a secure and loosely isolated environment. This secure application can run multiple containers simultaneously on the same host. Docker separates our application from its setup, which helps us in reducing the development time and deploying our project more quickly. It means that we can reduce the gap between the writing and running of code in the development phase. While using the docker application, we can use the same Linux kernel, like that of the system on which the application is running. Hence, we can run more containers on our system hardware as compared to the virtual machine.

Chapter 6

System Testing and Evaluation

In this chapter, we will discuss the system testing and evaluation.

6.1 Introduction

System testing is a level of testing that validates the complete and fully integrated software product [27]. System testing is mainly a series of different tests to ensure the proper working of our product and evaluate system specifications. In this part of the development our aim is to verify whether our project is working up to our needs or not. We do this testing to check for errors and to find out where fault and failures lie. Error is basically a mistake by the programmer that leads to a fault. A fault is a defect or a bug that further leads to failure. A failure occurs when the given requirements are not fulfilled. The following figure shows the relation between error, fault, and failure in system testing.

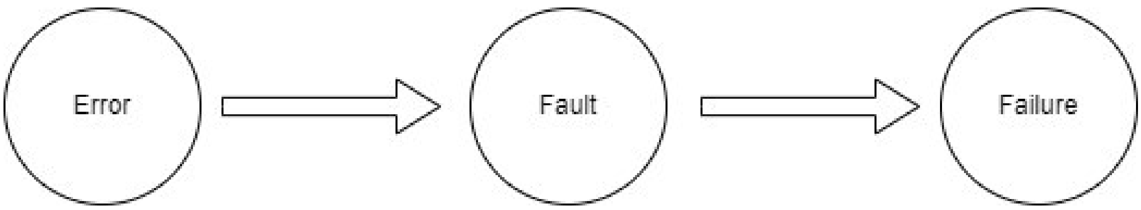

Figure 6.1: Relation between error, fault, and failure

6.2 Reasons for Testing

Testing is performed to ensure two very important tasks in the world of software development. They are verification and validation.

6.2.1 Verification

Verification is the testing of items to ensure that whether the software is in accordance with the specifications or not. It falls under static testing.

6.2.2 Validation

Validation is the evaluation done to determine that the specified requirements are satisfied. In other words, it is simply the testing to check that the software meets the customer expectations.

6.3 System Testing

6.3.1 Graphical User Interface Testing

GUI testing is an important component is any developed project. A good GUI allows the user to have a smooth interaction with the system using graphical components such as icons and buttons. For this purpose, we tested our screens, and all the feature's buttons, images, icons, and camera. The purpose of using graphical user interface was to present such a way that may allow the farmers to understand the results and use it without facing difficulties.

6.3.2 Usability Testing

Usability testing was carried out to ensure that the product is of exceptional quality and is highly user friendly so that there are no usability issues faced. Our developed system is very easy to use, as it requires very little user interactions. It was designed in such a simple way that the user with minimum technical knowledge can also use it. Our developed system was tested many times to check for any usability issues by loading pictures many times and checking for weeds. The interaction with the system was considered smooth with no issues found.

6.3.3 Compatibility Testing

Compatibility testing is done to check compatibility of our developed system. Essentially, to test that our developed system works fine on different browsers and fulfills user expectations.

6.3.4 Software Performance Testing

We have used software performance testing to access response time, resource usage and stability of our developed system. According to the tests our system is quite responsive and smooth. It takes a little longer once the images are being loaded but later the time decreases.

6.3.5 Exception Handling

There are two exceptions in our developed system. First is that the image should be clear and the second is that weeds should not be hidden under crops.

6.4 Test Cases

6.4.1 Startup Test Case

System starts correctly when launched.

Test case	
Test case ID	01
Description	Test the starting of system
Applicable for	Run on all browser systems
Requirement	REQ01
Initial condition	none
Step	Task and its expected results
1	Open web application
2	Verify startup on multiple systems
3	Verify the proper display of main screen
Status	Pass

Table 6.1: Test case 01.

6.4.2 Image Loading Test Case

To ensure that image loads successfully.

Test case	
Test case ID	02
Description	Test image loading.
Applicable for	Run on all browser systems.
Requirements	REQ02
Initial conditions	Startup test case successful.
Step	Tasks and expected result
1	Select option for uploading image.
2	Verify that it lets you select the image.
3	Verify that it is being uploaded.
Status	Pass

Table 6.2: Test case 02.

6.4.3 Weed Detection Test Case

To ensure that weeds are detected successfully.

Test case	
Test case ID	03
Description	Test weed detection
Applicable for	Run on all browser systems
Requirements	REQ03
Initial conditions	Image loaded successfully
Step	Tasks and expected result
1	Image is processing
2	Verify that detected weed is being displayed on the screen.
3	Verify that weed to crop ratio result is generated.
Status	Pass

Table 6.3: Test Case 03.

6.4.4 Report Generation Test Case

To ensure that report is generated successfully.

Test case	
Test case ID	04
Description	Test report generation
Applicable for	All browser systems
Requirements	REQ04
Initial conditions	Result generated successfully.
Step	Tasks and expected result
1	Select option for report generating.
2	Verify that it lets you send report.
3	Verify that report is being received via email.
Status	Pass

Table 6.4: Test Case 04.

6.5 Dataset

That dataset we used is Ronin_open_db. Our dataset contains images of weeds and crops captured in different light conditions. It contains around 1200 images of crop and weed in various environments closed, open and different lighting's we divided our data set into 85% for training 10% for validation 5% for testing.

6.5.1 Testing for Weed Under Natural Light

We used images from our dataset to test our developed system and its accuracy of weed detection in natural light. The detection was successful. Most of the pictures form the dataset are taken in natural light in natural light the results were very accurate.

Figure 6.2: Shows input image and detected weed under natural light.

6.5.2 Testing for Weed Under Different Lighting

We used images from our dataset to test our developed system and its accuracy of weed detection in Different Lighting. The detection was successful.

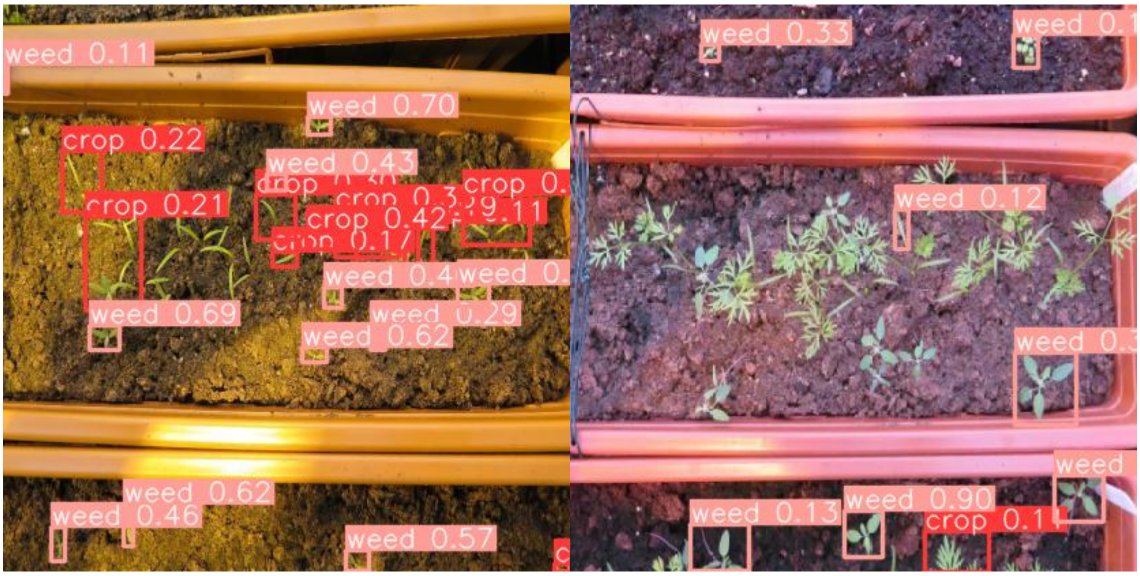

Figure 6.3: Weed Under Different Lighting Conditions.

6.5.3 Batch Testing

We used images from our dataset to test our developed system and its accuracy of weed detection in batches.

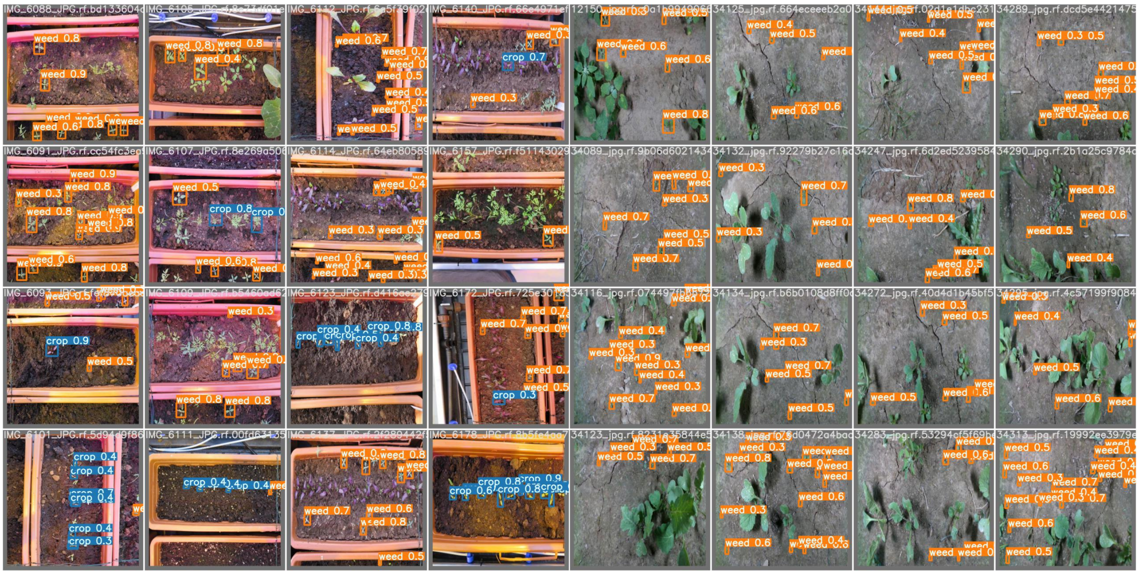

Figure 6.4: Batch test

6.6 Evaluation

The following figure shows the evaluation of trained model.

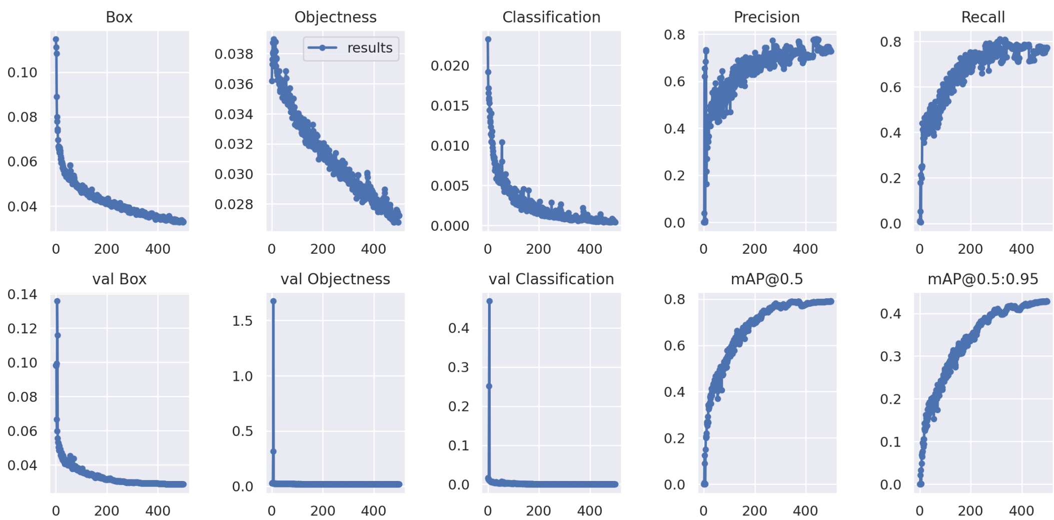

Figure 6.5: Shows graphical representation of evaluation of trained model.

6.6.1 Confusion Matrix

Confusion matrix is a metric to measure performance and accuracy of classification algorithm. It uses recall, precision, accuracy and visualises them in a manner which can provide values for true positives and false negatives etc.

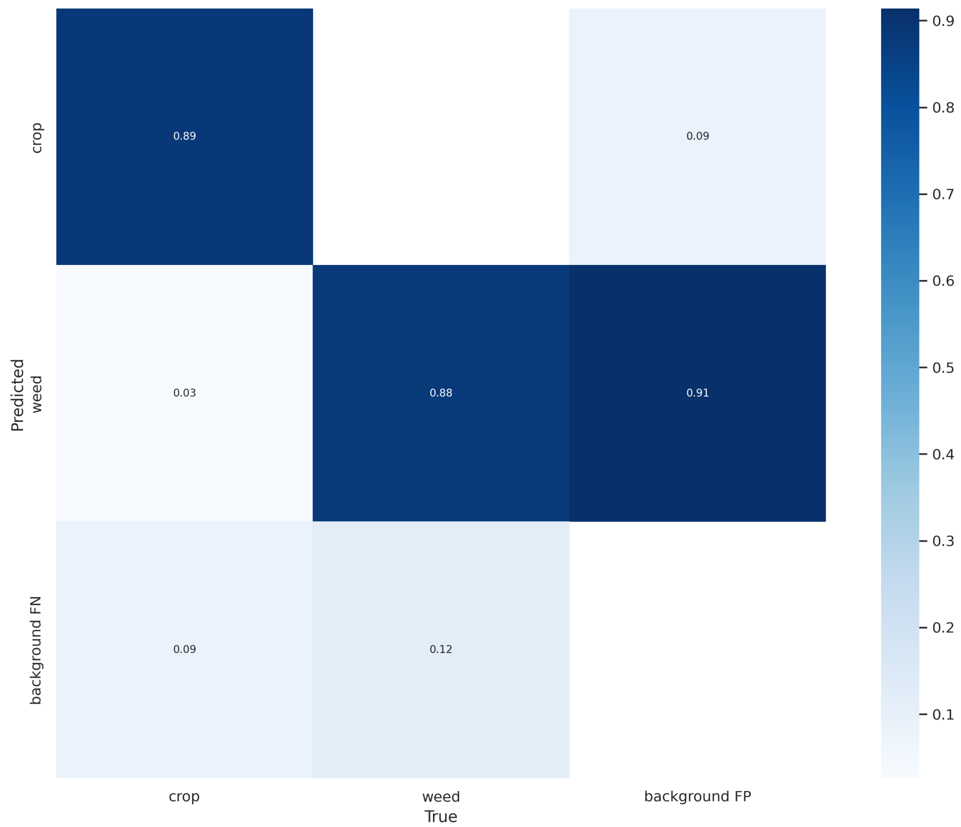

Figure 6.6: Confusion Matrix

6.6.2 Mean Average Precision

Mean average precision is a benchmark for object detection algorithms. It is average precision of all the classes as shown in the equation. It is denoted as mAp@[0.5 ,0.95] which indicates mean average precision over different intersection over union thresholds.

$$mAP = 1/N \sum_{i=1}^{N} AP_i$$

Class	Precision	Recall	mAP@.5
All	0.729	0.772	0.789
Crop	0.754	0.827	0.845
Weed	0.704	0.718	0.733

Table 6.5: Mean average precision of different classes

6.6.3 F1 Score

F1 score is a metric to measure performance using only precision recall and confidence. It combines two factors both precision and recall into one. It is used to find most optimal confidence score so that if any bounding box with the confidence score more than threshold is considered a detection.

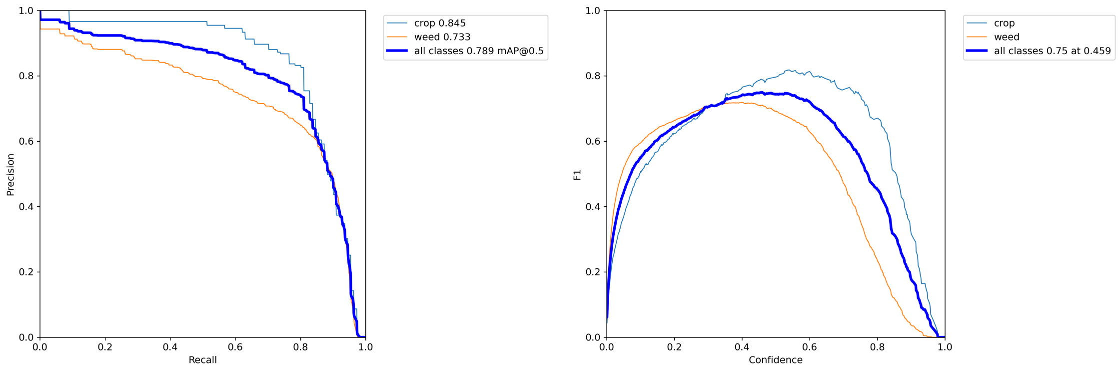

Figure 6.7: PR & F1 curve

Chapter 7

Conclusions

The identification of weeds and pests in natural environment is a crucial task and is of great importance to crop yield. Fast and efficient detection requires improved computational methods. We have built a system that will prove significant in this regard because of higher accuracy compared to the existing systems. In this system we have developed a method for weed detection and pest detection using image processing for crop health monitoring system based on yolov5. We can detect and separate out the unwanted plants from the useful crop plants and timely identify pests' infestation. The reason for developing such a system is to bring automation in this field which may help in precision agriculture.

Our idea is to help the local communities which are lagging due to the use of old traditional methods by bringing technology to improve their yield. With a rise in the use of IOT devices in the recent years it has become the need to link technology with agriculture to compete with the world. Loss of crop yield in the past few years stresses on the need to update our outdated methods used in agriculture. Our system provides on such solution by making use of the deep learning for the efficient and accurate detection of weeds and pests. It gives user the freedom to detect weeds or pests. It also generates a report showing the results of detected weeds and pests. We used email system to notify the crop manager regarding the current situation of the field.

There were limited online resources and research paper available about CNN- based weed detection system. One of the most difficult tasks in the making of this project was finding the required dataset which contained both crops and weeds in a single image. Moreover, training and implementing the system was a bit challenging but, in the end, we successfully completed our project and is working competently.

Our system can be used by any agriculture related company. They can provide timely reports to the farmers and in return get paid for each report generated. Our system is generic and can be integrated into any other mobile, web or desktop application because we used APIs such as object detection API for this job. The methodology employed in our system can be extended to other similar problems with very little adaptations.

7.1 Improvements for the Future

As far as future work is concerned it would be interesting to calculate accuracy and precision with images that are more complex and contains all sizes of crops and weeds that too in clusters and different sort of images with pests. There are also some features that can be later added to improve the functionality of our system.

1. A countdown or a timer that tells the time remaining in the crops to completely grow.
2. A predictor for the right time to water the crops.
3. We have used a dataset but to get more accurate results we can make our own custom dataset containing images taken in different seasons, different lightings, and different crops.
4. Weed recognition and labelling can also be added.
5. Extra modules such as crop diagnosis and rain prediction can be added.
6. Performance of application can be tuned using more advance technology.

Appendix A

User Manual

A.1 Introduction

User manual guide users about the way application can be used. We have provided description about the purpose of the screens. The screens of our application "Crop health monitoring system" are as follows:

A.2 Landing page

When user opens the URL for our application it is directed towards this landing page which has only one button which says "Try it out" which is referring towards weed detection page.

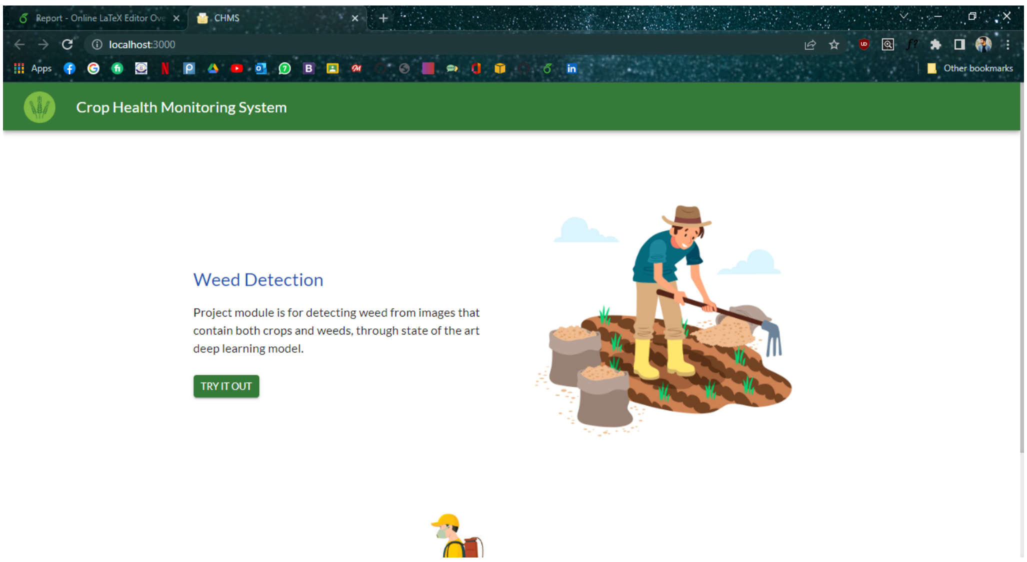

Figure A.1: Graphical user interface of landing page

A.3 Weed Detection Page

User is directed towards weed detection page then again another similar button upload picture for input.

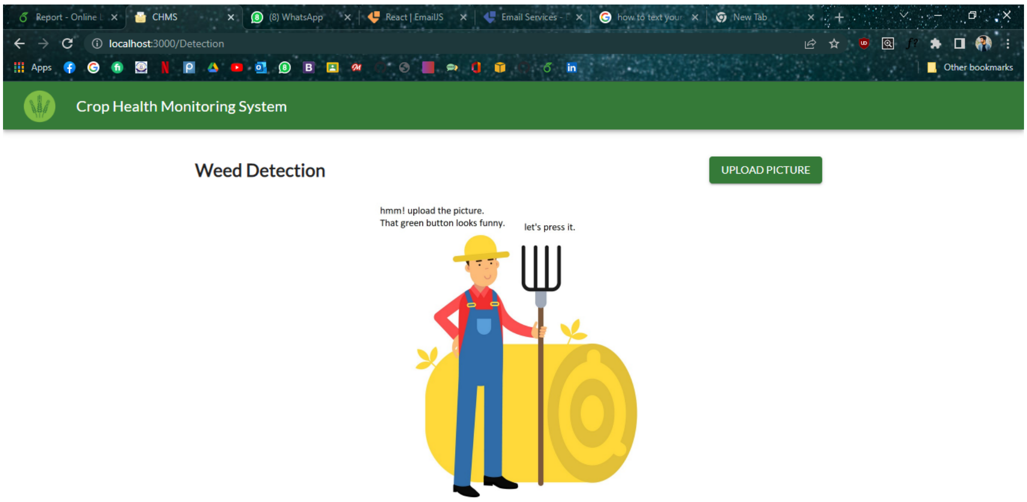

Figure A.2: Graphical user interface of weed detection page

After giving it an input this screen shows output and another button appears "Send report" this button notifies crop manger via email about the current situation of the field.

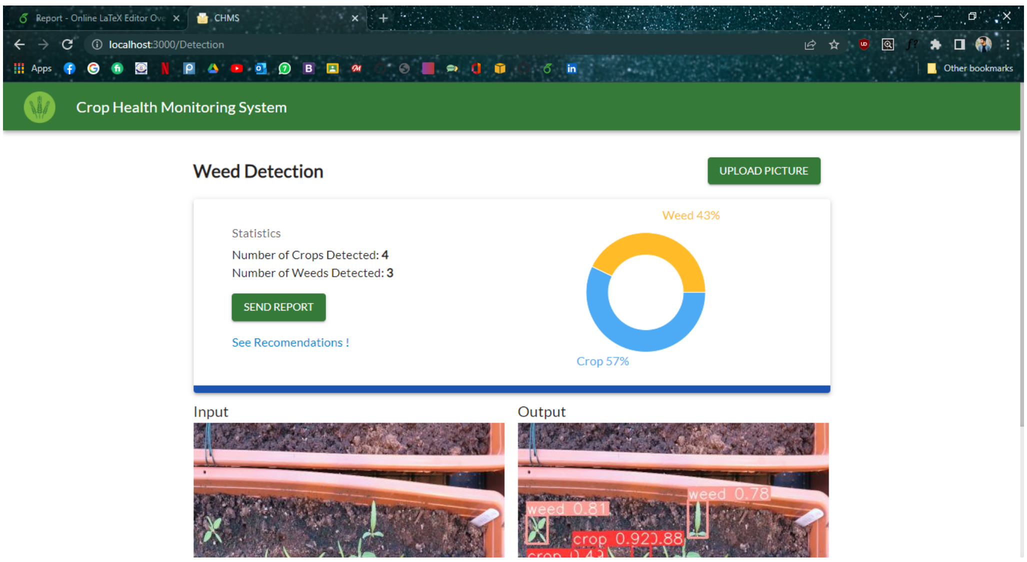

Figure A.3: Output of weed detection page

A.4 View Recommendation

On the output card there is a link "See Recommendation !" After clicking this link user is redirected to another screen.

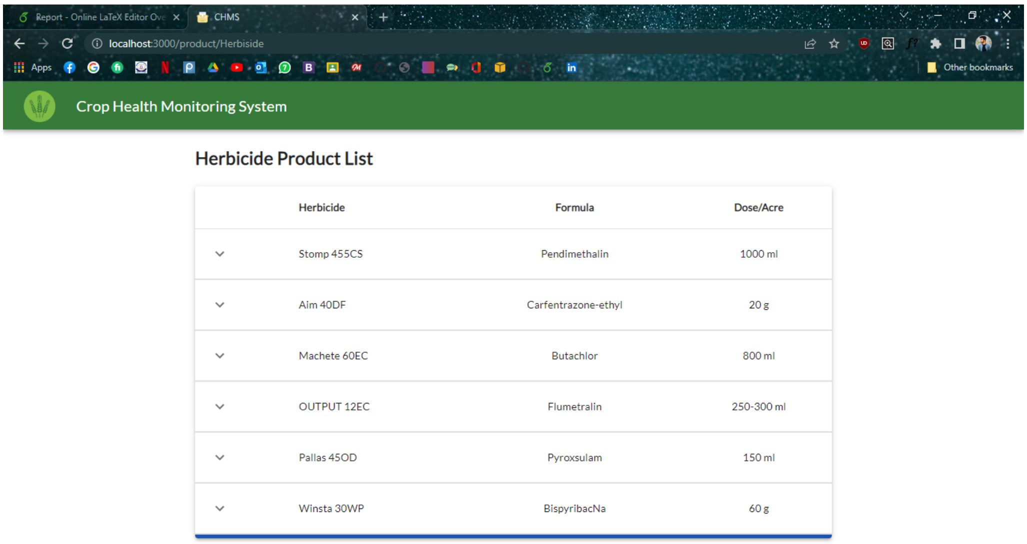

Figure A.4: Graphical user interface of recommendation page

After viewing recommendations list user can view detailed description of the product and in which crops it is used. Manager or an expert can then prescribe a product which can then shared via email.

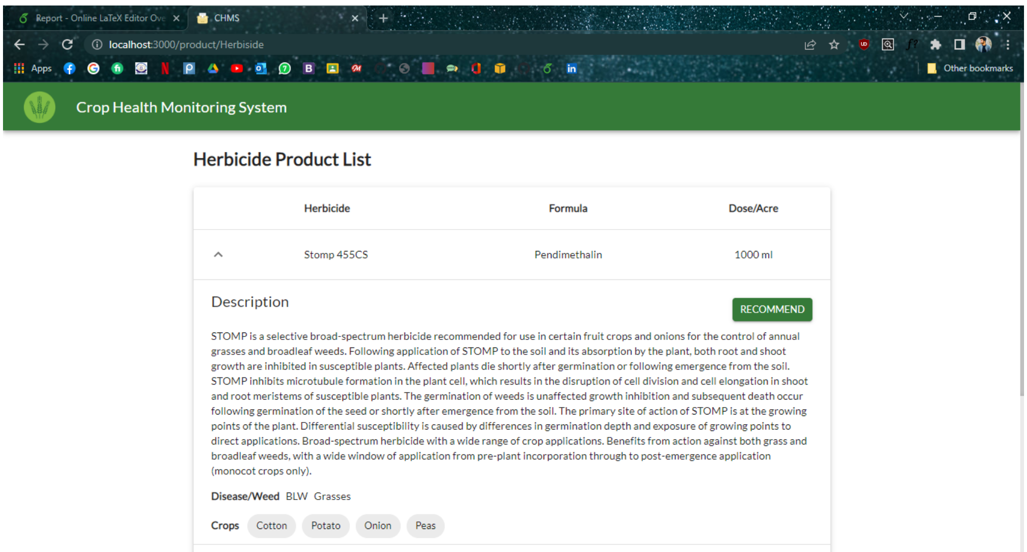

Figure A.5: Output of weed detection page

A.5 View Email

This screen shows the output of an email sent by the system to the manager. According to a set value of 40% threshold is crossed by the weed objects an email is sent notifying manager to take actions to prevent further growth of weed. if weed is less than 40% an email is sent notifying manager no actions are needed. In the second email crop recommendations are sent by manager.

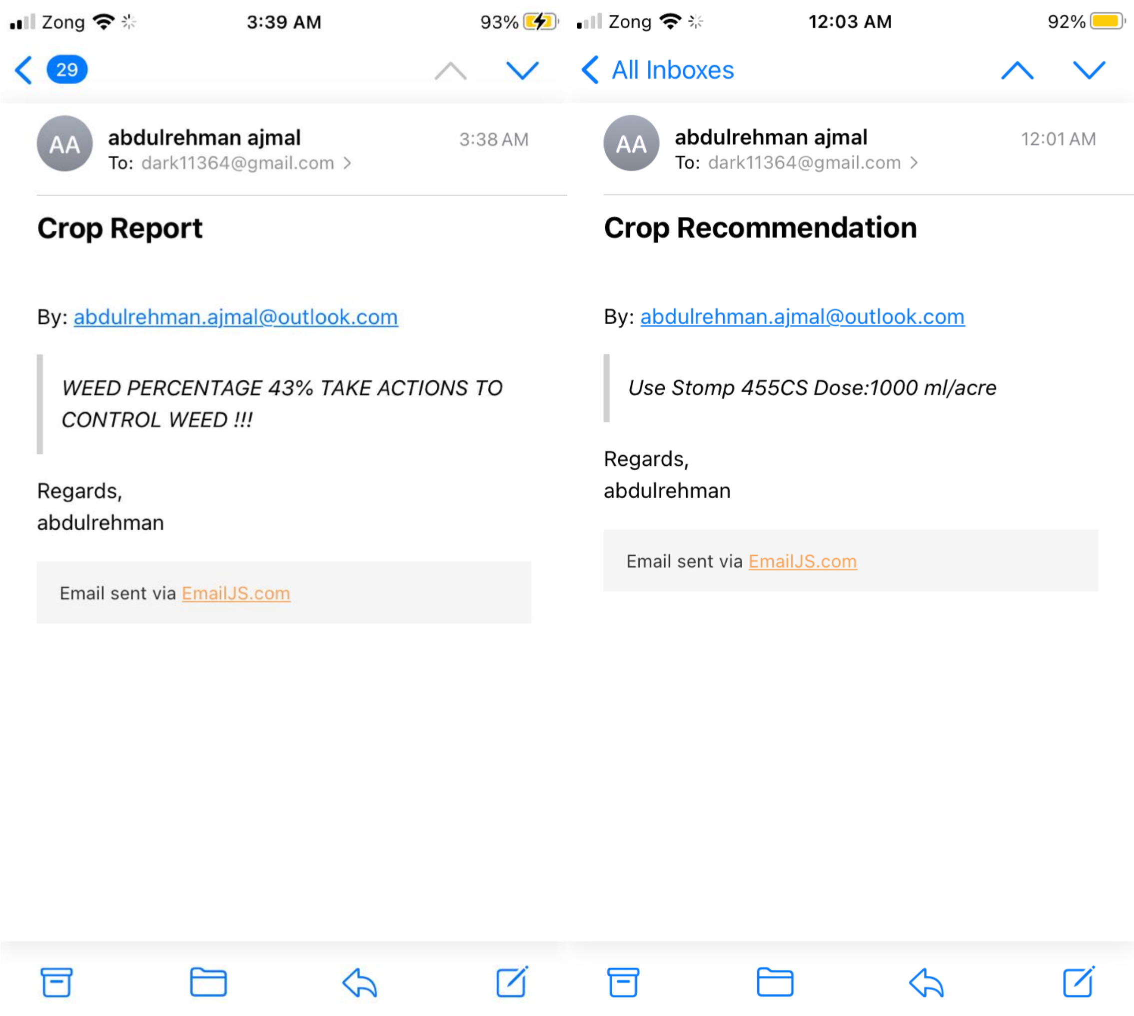

Figure A.6: Shows emails sent by the system

A.6 API Documentation

This screen shows the Swagger-UI of the api provided by fastapi framework. Admins or development teams can use it for basic functionality and testing.

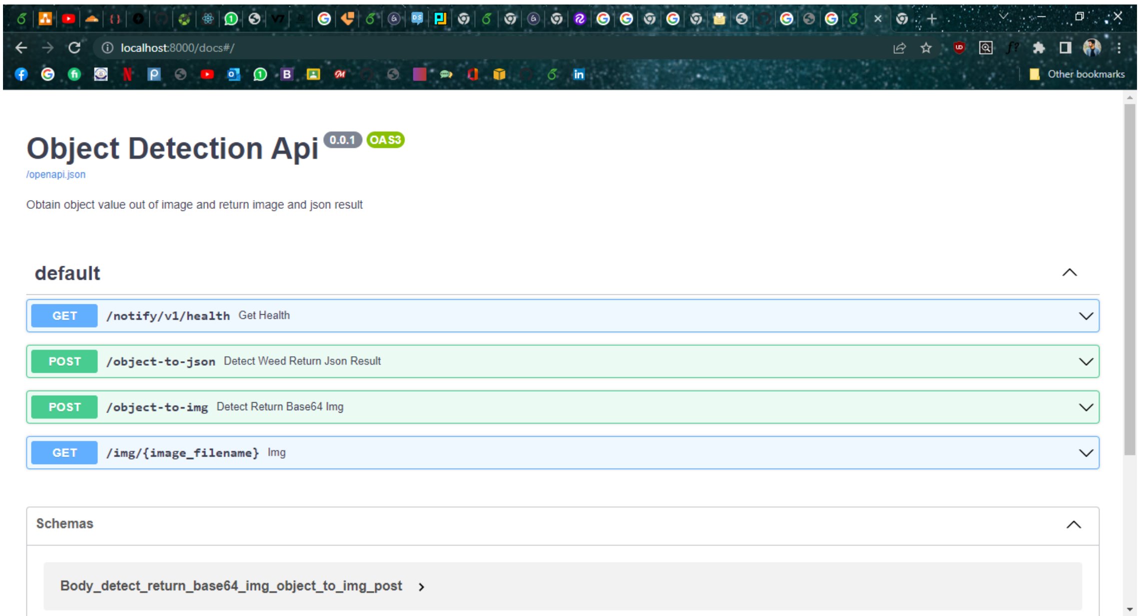

Figure A.7: Shows API's user interface

A.7 How to Use the Code

In this section it is described how to deploy the source code to a local server. Open terminal and follow these commands:

1. Clone this Repository on local machine.
   `https://github.com/Pineapple-1/weed-detection`
2. cd ./client/
3. npm install && start
4. cd ./yolov5-fastapi/
5. pip install -r requirements.txt
6. uvicorn main:app –host 0.0.0.0 –port 8000

These commands are only for the first time to setup all the packages and libraries necessary for the project. After that use only "npm start" to start the client and "uvicorn main:app –host 0.0.0.0 –port 8000" to start the server.

References

[1] Shah Jahan, Umed Ali Leghari, Ghulam Laghari, Mahmooda Buriro, and Farooque Soomro. An overview on various weed control practices affecting crop yield. *Journal of Chemical, Biological and Physical Sciences*, vol.6, N.1:59–69, 11 2015. Cited on pp. xi and 3.

[2] Sylvia Fronczak. Iterative and incremental method. Cited on pp. xi and 33.

[3] Zhangnan Wu, Yajun Chen, Bo Zhao, Xiaobing Kang, and Yuanyuan Ding. Review of weed detection methods based on computer vision. *Sensors*, 21(11), 2021. Cited on p. 1.

[4] Dawn. Weeds and their ill-effects on main crops. Cited on pp. 1 and 2.

[5] Amar Matloob, Muhammad Ehsan Safdar, Tasawer Abbas, Farhena Aslam, Abdul Khaliq, Asif Tanveer, Abdul Rehman, and Asim Raza Chadhar. Challenges and prospects for weed management in pakistan: A review. *Crop Protection*, 134:104724, 2020. Cited on p. 1.

[6] I Gnanavel et al. Eco-friendly weed control options for sustainable agriculture. *Science International (Dubai)*, 3(2):37–47, 2015. Cited on p. 3.

[7] Vi Nguyen Thanh Le, Beniamin Apopei, and Kamal Alameh. Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods. *Information processing in agriculture*, 6(1):116–131, 2019. Cited on pp. 7 and 12.

[8] Yajun Chen, Zhangnan Wu, Bo Zhao, Caixia Fan, and Shuwei Shi. Weed and corn seedling detection in field based on multi feature fusion and support vector machine. *Sensors*, 21(1):212, 2020. Cited on pp. 7 and 13.

[9] Ma Yuan, Feng Quan, Yang Mei, and LI Miaoqi. Detection of wine grape leaves based on hog. *Computer Engineering and Applications*, 52(15):158–161, 2016. Cited on pp. 7 and 13.

[10] Adel Bakhshipour and Abdolabbas Jafari. Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Computers and Electronics in Agriculture*, 145:153–160, 2018. Cited on pp. 7 and 13.

[11] Yanjie Li, Mahmoud Al-Sarayreh, Kenji Irie, Deborah Hackell, Graeme Bourdot, Marlon M Reis, and Kioumars Ghamkhar. Identification of weeds based on hyperspectral imaging and machine learning. *Frontiers in Plant Science*, 11:2324, 2021. Cited on pp. 8, 9, and 13.

[12] Ana I De Castro, Jorge Torres-Sánchez, Jose M Peña, Francisco M Jiménez-Brenes, Ovidiu Csillik, and Francisca López-Granados. An automatic random forest-obia algorithm for early weed mapping between and within crop rows using uav imagery. *Remote Sensing*, 10(2):285, 2018. Cited on pp. 8 and 13.

[13] Sadia Murawwat, Armish Qureshi, Saleha Ahmad, and Yousaira Shahid. Weed detection using svms. *Engineering, Technology & Applied Science Research*, 8(1):2412–2416, 2018. Cited on pp. 8 and 14.

[14] Jialin Yu, Arnold W. Schumann, Zhe Cao, Shaun M. Sharpe, and Nathan S. Boyd. Weed detection in perennial ryegrass with deep learning convolutional neural network. *Frontiers in Plant Science*, 10:1422, 2019. Cited on p. 10.

[15] Mahmoud Abdulsalam and Nabil Aouf. Deep weed detector/classifier network for precision agriculture. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 1087–1092, 2020. Cited on p. 10.

[16] Kavir Osorio, Andrés Puerto, Cesar Pedraza, David Jamaica, and Leonardo Rodríguez. A deep learning approach for weed detection in lettuce crops using multispectral images. *AgriEngineering*, 2(3):471–488, 2020. Cited on p. 10.

[17] C Thirumarai Selvi, RS Sankara Subramanian, and R Ramachandran. Weed detection in agricultural fields using deep learning process. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1470–1473. IEEE, 2021. Cited on pp. 11 and 15.

[18] Junfeng Gao, Andrew P French, Michael P Pound, Yong He, Tony P Pridmore, and Jan G Pieters. Deep convolutional neural networks for image-based convolvulus sepium detection in sugar beet fields. *Plant Methods*, 16(1):1–12, 2020. Cited on pp. 11 and 15.

[19] Xiaojun Jin, Jun Che, and Yong Chen. Weed identification using deep learning and image processing in vegetable plantation. *IEEE Access*, 9:10940–10950, 2021. Cited on pp. 12 and 15.

[20] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, and Marcelo Theophilo Folhes. Weed detection in soybean crops using convnets. *Computers and Electronics in Agriculture*, 143:314–324, 2017. Cited on p. 14.

[21] Jialin Yu, Arnold W Schumann, Zhe Cao, Shaun M Sharpe, and Nathan S Boyd. Weed detection in perennial ryegrass with deep learning convolutional neural network. *Frontiers in plant science*, 10:1422, 2019. Cited on p. 14.

[22] Mahmoud Abdulsalam and Nabil Aouf. Deep weed detector/classifier network for precision agriculture. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 1087–1092. IEEE, 2020. Cited on p. 14.

[23] Kavir Osorio, Andrs Puerto, Cesar Pedraza, David Jamaica, and Leonardo Rodrguez. A deep learning approach for weed detection in lettuce crops using multispectral images. *AgriEngineering*, 2(3):471–488, 2020. Cited on p. 14.

[24] Luiz Carlos M Junior and José Alfredo C Ulson. Real time weed detection using computer vision and deep learning. In *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1131–1137. IEEE, 2021. Cited on p. 15.

[25] Renjie Xu, Haifeng Lin, Kangjie Lu, Lin Cao, and Yunfei Liu. A forest fire detection system based on ensemble learning. *Forests*, 12:217, 02 2021. Cited on p. 37.

[26] Kaspars Sudars, Janis Jasko, Ivars Namatevs, Liva Ozola, and Niks Badaukis. Dataset of annotated food crops and weed images for robotic computer vision control. *Data in brief*, 31:105833, 2020. Cited on p. 39.

[27] Thomas Hamilton. What is system testing? types & definition with example. Cited on p. 43.