A

Major Project

On

E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

POKKILI SAMPATH KUMAR(197R1A05P1)

MUTHE AKHIL KUMAR (197R1A05N4)

REDDY RUDHRAMSH REDDY (197R1A05P3)

Under the Guidance of

G. KALPANA DEVI

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING"** being submitted by **P.SAMPATH KUMAR(197R1A05P1)**, **M.AKHIL KUMAR(197R1A05N4), R. RUDHRAMSH REDDY(197R1A05P3)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

G. Kalpana Devi
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **G. Kalpana Devi,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. T . Subha Mastan Rao & J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

POKKILI SAMPATH KUMAR (197R1A05P1)
MUTHE AKHIL KUMAR (197R1A05N4)
REDDY RUDHRAMSH REDDY (197R1A05P3)

ABSTRACT

This project presents an agricultural aid application, developed and designed, to help farmers by utilizing Image Processing, Machine Learning and Deep Learning concepts. Our application provides features such as early detection of plant disease, implemented using various approaches. After evaluation, results showed that Convolutional Neural Network was per forming better for plant disease detection with an high accuracy. It further helps the farmer to forecast the weather to decide the right time for agricultural activities like harvesting and plucking. To avoid reoccurrence of disease due to loss in soil minerals, a crop specific fertilizer calculator is incorporated which can calculate the amount of urea, diammonium phosphate and muriate of potash required for a given area.

This project showcases an agricultural aid app that was built and designed to assist farmers by employing Image Processing, Machine Learning, and Deep Learning. Features like early detection of plant disease are available in our application and are implemented in a number of ways. It was determined that Convolutional Neural Network was superior for detecting plant diseases with a high degree of accuracy. The farmer can use the weather forecast to plan out agricultural tasks like harvesting and plucking at the optimal time. A crop-specific fertilizer calculator is in the works to determine how much urea, diammonium phosphate, and muriate of potash should be applied to a given area to prevent the recurrence of disease caused by depleted soil minerals.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The proposed methodology is used for the precise detection of disease in crops. Which can provide controlled fertilization to farmers. Accurate identification of disease also helps farmers to identify the infection and do relatively controlled fertilization to avoid any future crop failures.

1.2 PROJECT PURPOSE

This project has been developed for the main purpose of detecting the spam and fraud accounts on social media platform Twitter. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. The only purpose of this project is just to save the users time and their confidentiality on Twitter. We are hopeful that this project will be a useful resource for Twitter spam detection on a single platform. By this project we can reduce the spammers and fake users by this negativity of social media will reduce and also the reputations of those social media will increase in a positive manner.

1.3 PROJECT FEATURES

Our application provides features such as early detection of plant disease, implemented using various approaches. After evaluation, results showed that Convolutional Neural Network was performing better for plant disease detection with an accuracy of 97.94% at 20 epochs.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst hasa firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Plant diseases can affect vast produce of crops posing a major menace to food security as well as leading to major losses to farmers. An extensive review of existing research was conducted by us on this domain [5] and in an effort to help farmers overcome this problem, we have designed an android application, Agricultural Aid which utilizes machine learning to provide plant disease detection. This detection is combined with an android application which provides features like weather forecast of up to 7 days, fertilizer calculator and language translation in up to 4 languages which has been implemented and integrated using Android Studio and its APIs. For disease classification, we followed two approaches: Image Processing with Machine Learning and Deep Learning models.

2.2 EXISTING SYSTEM

Image Processing is a popular first step in the process of plant disease detection and often includes multi-step processes to achieve processed, ROI centric input images for further classification. In this approach, various Image Processing techniques were used on the input image to get a final output image which would mark the infected area and also calculate the percentage of area infected in the leaf. The major advantage observed in this approach was that this approach eliminated the need to extract the leaf and place it on a black background during image capture for the algorithm to work. In real time scenario, the infected leaf would be present amongst a cluster of mixed crops and this approach was able to segregate the infected leaf from the healthy ones in an input image.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- Accuracy is Low.
- This multistep algorithm is that it is not optimized and no better background elimination results.

2.3 PROPOSED SYSTEM

In the Deep Learning Approach, we decided to take a subset of the Plant Village dataset along with the cotton dataset to train and test the CNN model. The input image is fed into this model, which initially took a portion of the Plant Village and Cotton Dataset, with a training -validation split of 70-30, therefore getting 4200 images for training and 1800 images for validation. A CNN (Convolutional Neural Network) is a deep learning model that takes inputs which are assigned weights depending on various features. CNN is a widely used neural network for image-based datasets.

Our CNN model consists of 4 main convolutional layers with 32, 64, 128, 128 filters consecutively, each followed by a ReLU activation function, max pooling and dropout layer. This set of convolutional layers is followed by a flatten and then a dense layer which is finally followed by the softmax activation function that tells us which class has the maximum probability.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Accuracy is Very high.
- The reason for choosing CNN algorithm is that it is more optimized and thus gives better background elimination results than the previous multistep approach.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This isto ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminaryinvestigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual costto spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Core i3 or higher
- Hard Disk : 4 GB or higher
- Memory : 5GB and higher
- Internet Connection : Broadband or higher
- GPU : minimum of 2GB - VRAM

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.

The following are some software requirements.

- Operating Systems : Windows 10 or higher / Ubuntu 16.04 or higher
- Programming Language : Python 3.7 or higher
- Python IDE : Anaconda, PyCharm or Spyder
- Libraries : NumPy, Pandas, Scikit-learn, Matplotlib
- Twitter dataset : JavaScript Object Notation (JSON) format

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure how text is converted into code and speech is converted in to text.

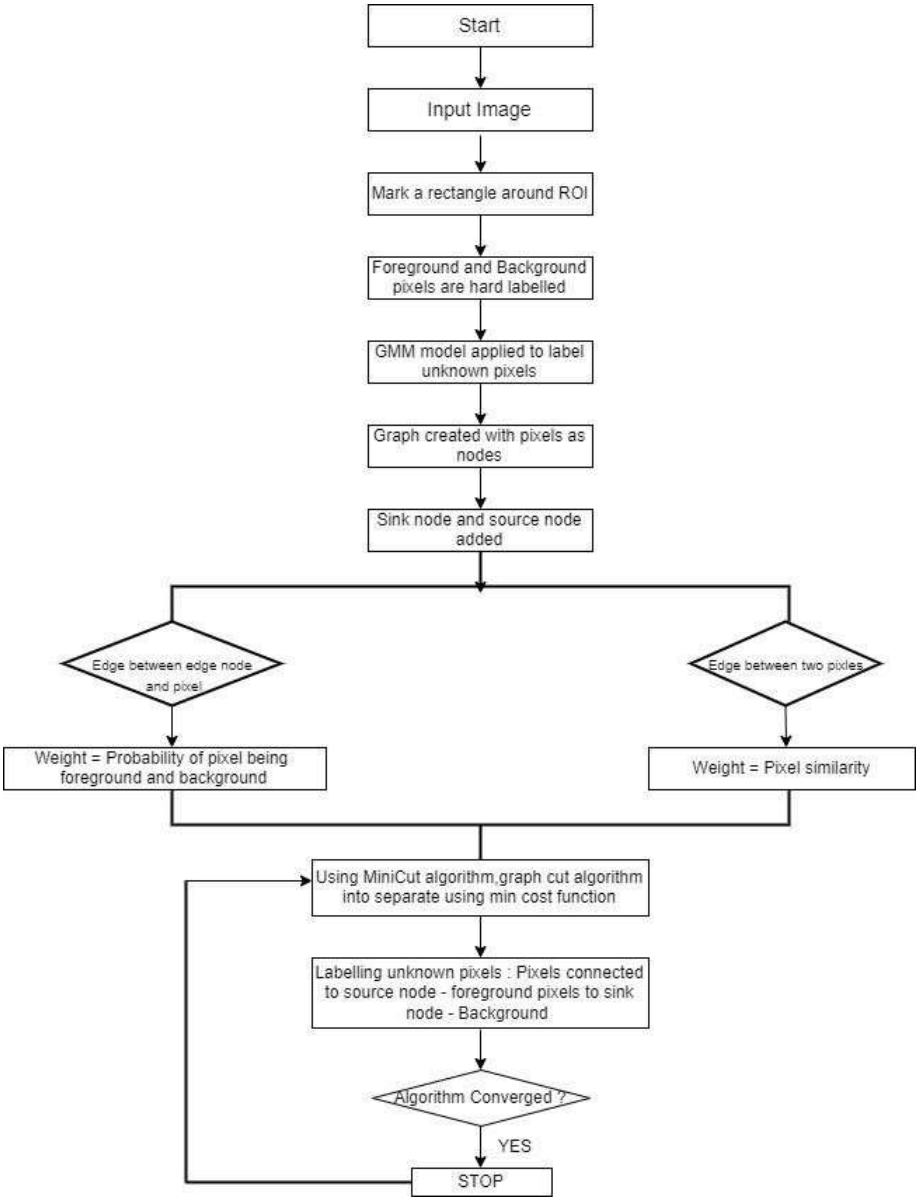


Figure 3.1: Project Architecture for E-Agri Kit Agricultural Aid Using Deep Learning

3.2 DESCRIPTION

1) After selecting the image for extraction, the user is required to mark out a rectangle selecting the region/object of interest (ROI) where everything outside the rectangle is taken as background.

2) The computer labels the pixels as “foreground” and “background” pixels which is also known as hard labelling.

3) After labelling, a Gaussian Mixture Model (GMM) is applied on the foreground and background. GMM learns and tries to label unknown pixels inside the rectangle as either probably foreground or background according to hard-labelled pixels and color statistic which is similar to clustering.

4) Using this pixel distribution, a graph is created where pixels are denoted as nodes. Additionally, 2 nodes are added, namely: Source node and Sink node such that every foreground pixel is connected to source node and every background pixel is connected to sink node.

5) The weight of the edges connecting pixels to source node is calculated as the probability of the pixel being a part of the foreground or the background.

6) The weight of the edges connecting two pixels is defined by pixel similarity such that if there is a small difference in pixel color the weight will be high.

7) Using a min-cut algorithm on the graph, the graph is cut into separate source node and sink node using minimum cost function where the cost function is sum of all the edges that are cut.

8) After the cut, nodes (pixels) connected to source node are labelled as foreground and nodes(pixels) connected to sink node becomes background.

3.3 MODULES

Modules Used in Project :-

3.3.1 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

3.3.2 Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

3.3.1.1 A powerful N-dimensional array object

3.3.1.2 Sophisticated (broadcasting) functions

3.3.1.3 Tools for integrating C/C++ and Fortran code

3.3.1.4 Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide varietyof databases.

3.3.3 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem.

3.3.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

3.3.5 Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

3.4 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

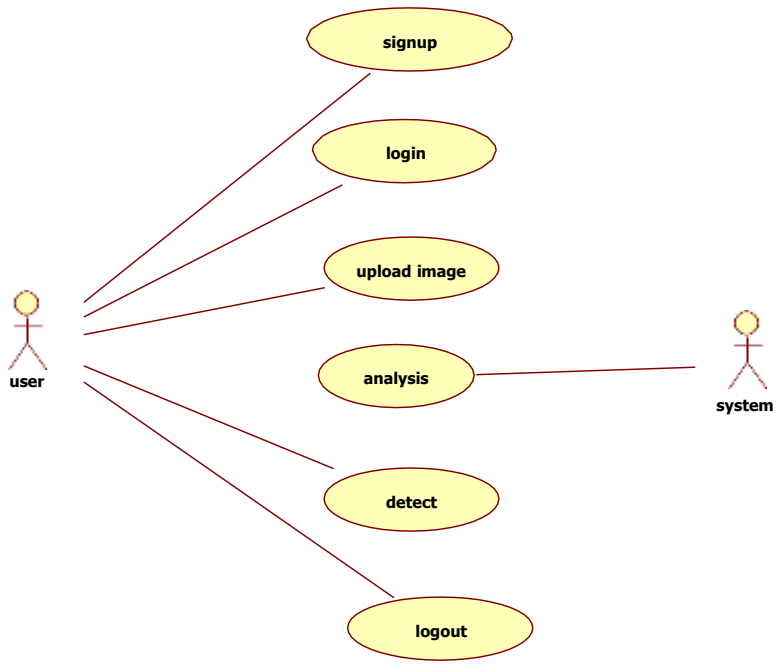


Figure 3.2: Use Case Diagram for E-Agri Kit Agricultural Aid Using Deep Learning

3.5 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (methods), and the relationships among objects.

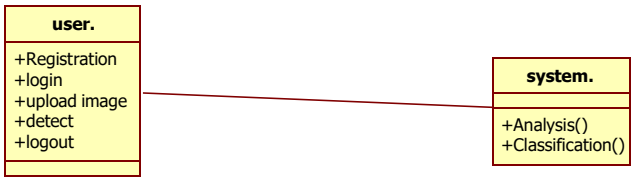


Figure 3.3: Class Diagram for E-Agri Kit Agricultural Aid Using Deep Learning

3.6 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

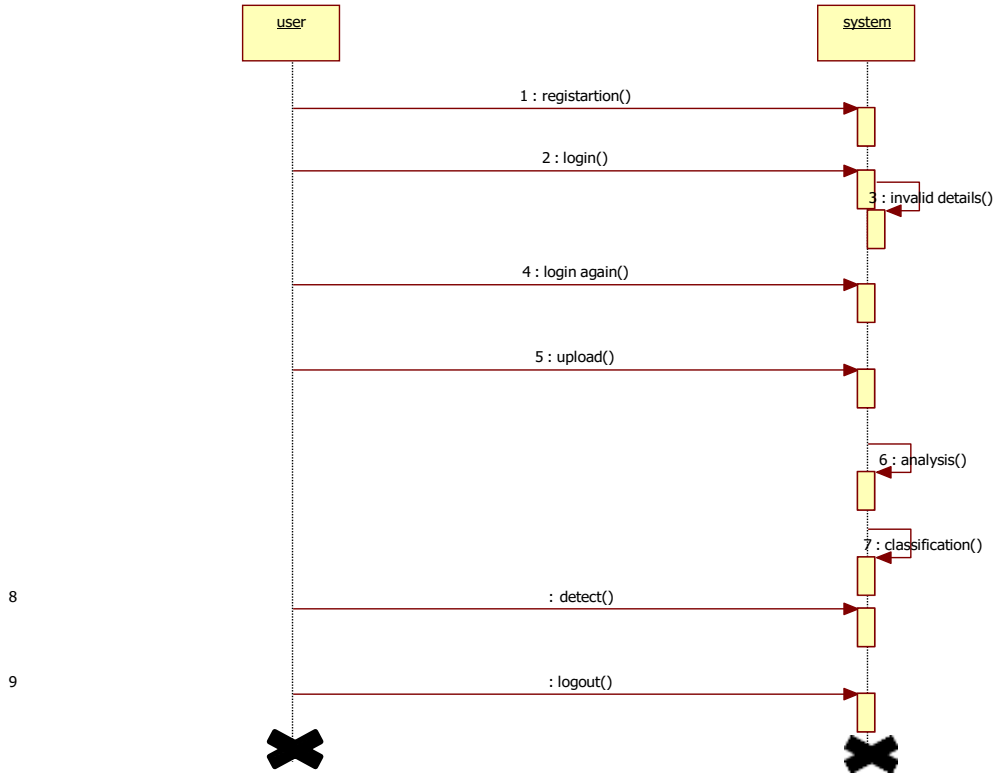


Figure 3.4: Sequence Diagram for E-Agri Kit Agricultural Aid Using Deep Learning

3.7 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can alsoinclude elements showing the flow of data between activities through one or more datastores.

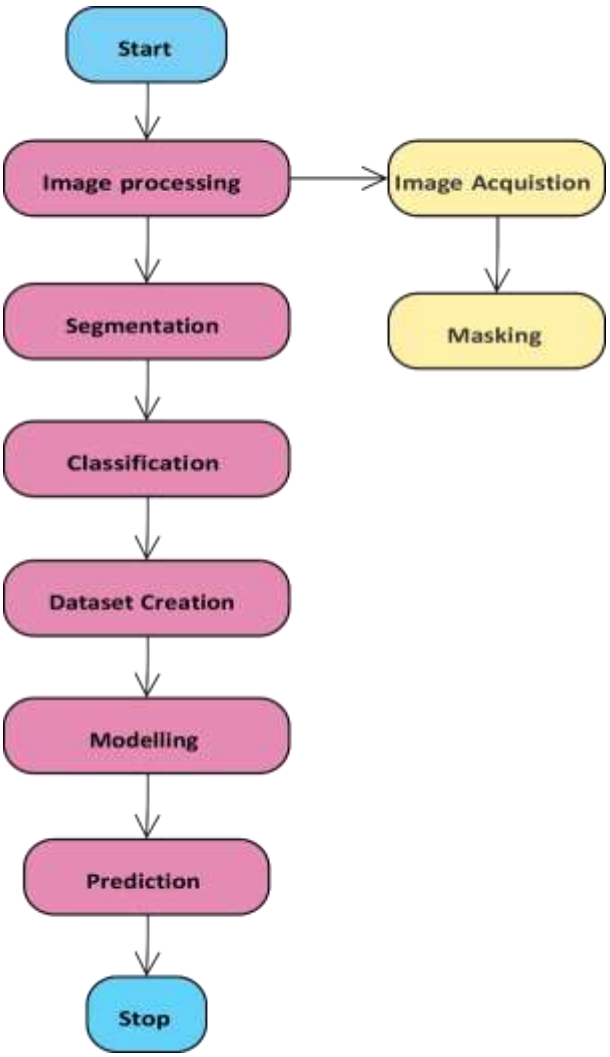


Figure 3.5: Activity Diagram for E-Agri Kit Agricultural Aid Using Deep Learning

4. IMPLEMENTATION

4.IMPLEMENTATION

4.1 SAMPLE CODE

```
from__future__import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import pandas as pd
import requests
import pickle
# Keras
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model
from keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
import sqlite3

# Define a flask app
app = Flask(_name_)

UPLOAD_FOLDER = 'static/uploads/'

# allow files of a specific type
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

# function to check the file extension
def allowed_file(filename):
```

```
  return '.' in filename and \
         filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/')
def home():
                                return render_template('home.html')

@app.route('/logon')
def logon():
                                return render_template('signup.html')

@app.route('/login')
def login():
                                return render_template('signin.html')

@app.route("/signup")
def signup():

    username = request.args.get('user','')
    name = request.args.get('name','')
    email = request.args.get('email','')
    number = request.args.get('mobile','')
    password = request.args.get('password','')
    con = sqlite3.connect('signup.db')
    cur = con.cursor()
    cur.execute("insert into `info` (`user`,`email`, `password`,`mobile`,`name`)
VALUES (?, ?, ?, ?, ?)",(username,email,password,number,name))
    con.commit()
    con.close()
    return render_template("signin.html")

@app.route("/signin")
def signin():

    mail1 = request.args.get('user','')
    password1 = request.args.get('password','')


  con = sqlite3.connect('signup.db')
```

```
    cur = con.cursor()
    cur.execute("select `user`, `password` from info where `user` = ? AND `password` = ?",(mail1,password1,))
    data = cur.fetchone()

    if data == None:
        return render_template("signin.html")

    elif mail1 == 'admin' and password1 == 'admin':
        return render_template("index.html")

    elif mail1 == str(data[0]) and password1 == str(data[1]):
        return render_template("index.html")
    else:
        return render_template("signup.html")

@app.route('/index', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')



model_path2 = 'models/model_xception.h5' # load .h5 Model
classes2 = {0:"Bacteria",1:"Fungi",2:"Nematodes",3:"Normal",4:"Virus"}
CTS = load_model(model_path2)
from keras.preprocessing.image import load_img, img_to_array
def model_predict2(image_path,model):
    print("Predicted")
    image = load_img(image_path,target_size=(224,224))
    image = img_to_array(image)
    image = image/255
    image = np.expand_dims(image,axis=0)

    result = np.argmax(model.predict(image))


 print(result)
    #prediction = classes2[result]
```

```
    if result == 0:
        return "Bacteria", "result.html"
    elif result == 1:
        return "Fungi","result.html"
    elif result == 2:
        return "Nematodes","result.html"
    elif result == 3:
        return "Normal","result.html"
    elif result == 4:
        return "Virus","result.html"



@app.route('/predict2',methods=['GET','POST'])
def predict2():
    print("Entered")
    if request.method == 'POST':
        print("Entered here")
        file = request.files['file'] # fet input
        filename = file.filename
        print("@@ Input posted = ", filename)

        file_path = os.path.join(UPLOAD_FOLDER, filename)
        file.save(file_path)
        #filename1 = os.path.join(UPLOAD_FOLDER, filename)

        print("@@ Predicting class.....")
        pred, output_page = model_predict2(file_path,CTS)

        remdies = []

        if pred == 'Bacteria':



con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("select `label` from data2 where `message` = ?",(pred,))
```

```
            remdies = cur.fetchall()

        elif pred == 'Fungi':
            con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("select `label` from data2 where `message` = ?",(pred,))
            remdies = cur.fetchall()

        elif pred == 'Nematodes':
            con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("select `label` from data2 where `message` = ?",(pred,))
            remdies = cur.fetchall()

        elif pred == 'Virus':
            val = 'viruses'
            con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("select `label` from data2 where `message` = ?",(val,))
            remdies = cur.fetchall()

        else:
            pred = pred

        return render_template(output_page, pred_output = pred, remdy = remdies,
img_src=UPLOAD_FOLDER + file.filename)


    #this section is used by gunicorn to serve the app on Heroku

data1 = pd.read_csv("data/Crop_recommendation.csv", encoding= 'unicode_escape')

@app.route('/crop')


def crop():
    companies=sorted(data1['Rural Areas'].unique())

    return render_template('crop.html',companies=companies)
```

```
@app.route('/features')
def features():
    return render_template('features.html')

data = pd.read_csv("data/Crop_recommendation.csv", encoding= 'unicode_escape')
data = data[['N', 'P', 'K', 'temperature', 'humidity','ph','rainfall','Rural Areas','label']]

from sklearn import preprocessing

# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
data['label']= label_encoder.fit_transform(data['label'])
data['Rural Areas']= label_encoder.fit_transform(data['Rural Areas'])
data['label'].unique()

x = data.iloc[:, 0:8]
y = data.iloc[:,8]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33, random_state=42)
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier()
RF.fit(x_train, y_train)
predictions = RF.predict(x_test)

@app.route('/predict',methods=['POST'])
def predict():
    N = request.form['N']
    P = request.form['P']
    K = request.form['K']



T = request.form['T']
    H = request.form['H']
    P = request.form['P']
    R = request.form['R']
    R1 = request.form['R1']
```

```
    reg = request.form['1']

    if reg == 'Adilabad':
        reg1 = 1
    elif reg == 'Bhadradri':
        reg1 = 2
    elif reg == 'Bhadradri Kothagudem':
        reg1 = 3
    elif reg == 'Hyderabad':
        reg1 = 4
    elif reg == 'Jagtial':
        reg1 = 5
    elif reg == 'Kamareddy':
        reg1 = 6
    elif reg == 'Karimnagar':
        reg1 = 7
    elif reg == 'Khammam':
        reg1 = 8
    elif reg == 'Kothagudem':
        reg1 = 9
    elif reg == 'Mahabubnagar':
        reg1 = 10
    elif reg == 'Mancherial':
        reg1 = 11
    elif reg == 'Medak':
        reg1 = 12
    elif reg == 'Medchal':
        reg1 = 13
    elif reg == 'Nagarkurnool':
        reg1 = 14
    elif reg == 'Nalgonda':


reg1 = 15
    elif reg == 'Nirmalll':
        reg1 = 16
    elif reg == 'Nizamabad':
        reg1 = 17
    elif reg == 'Peddapalli':
```

```
        reg1 = 18
    elif reg == 'Rangareddy':
        reg1 = 19
    elif reg == 'Sangareddy':
        reg1 = 20
    elif reg == 'Siddipet':
        reg1 = 21
    elif reg == 'Suryapet':
        reg1 = 22
    elif reg == 'Vikarabad':
        reg1 = 23
    elif reg == 'wanaparthy':
        reg1 = 24
    elif reg == 'Warangal':
        reg1 = 25
    elif reg == 'Warangal':
        reg1 = 26
    elif reg == 'Yadadri Bhuvangiri':
        reg1 = 27

    Soil_composition_list = np.array([N,P,K,T,H,P,R,reg1]).reshape(1,8)
    print(Soil_composition_list)

    crop = RF.predict(Soil_composition_list)
    print(crop)


    outcome = crop[0]
    if outcome == 0:
        predicted_crop = 'Rice'


  crop_link = 'https://en.wikipedia.org/wiki/Rice'
        crop_img = '../static/styles/images/rice.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )
```

```
    elif outcome == 1:
        predicted_crop = 'Rice'
        crop_link = 'https://en.wikipedia.org/wiki/Maize'
        crop_img = '../static/styles/images/maize.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 2:
        predicted_crop = 'Chickpea'
        crop_link = 'https://en.wikipedia.org/wiki/Chickpea'
        crop_img = '../static/styles/images/chikpea.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )

    elif outcome == 3:
        predicted_crop = 'Kidney Beans'
        crop_link = 'https://en.wikipedia.org/wiki/Rajma'
        crop_img = '../static/styles/images/kideneybeans.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 4:


 predicted_crop = 'Pigeon Peas'
        crop_link = 'https://en.wikipedia.org/wiki/Pigeon_pea'
        crop_img = '../static/styles/images/pigeonpea.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )
```

```
    elif outcome == 5:
        predicted_crop = 'Mothbeans'
        crop_link = 'https://en.wikipedia.org/wiki/Vigna_acconitifolia'
        crop_img = '../static/styles/images/mothbeans.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 6:
        predicted_crop = 'Mung Bean'
        crop_link = 'https://en.wikipedia.org/wiki/Mung_bean'
        crop_img = '../static/styles/images/mungbeans.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )

    elif outcome ==  7:
        predicted_crop = 'Black Grams'
        crop_link = 'https://en.wikipedia.org/wiki/Vigna_mungo'
        crop_img = '../static/styles/images/blackgram.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )




elif outcome == 8:
        predicted_crop = 'Pomegranate'
        crop_link = 'https://en.wikipedia.org/wiki/Pomegranate'
        crop_img = '../static/styles/images/pomogranate.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )
```

```
    elif outcome == 9:
        predicted_crop = 'Banana'
        crop_link = 'https://en.wikipedia.org/wiki/Banana'
        crop_img = '../static/styles/images/banana.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 10:
        predicted_crop = 'Mango'
        crop_link = 'https://en.wikipedia.org/wiki/Mango'
        crop_img = '../static/styles/images/mango.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 11:
        predicted_crop = 'Grapes'
        crop_link = 'https://en.wikipedia.org/wiki/Grape'
        crop_img = '../static/styles/images/'




return render_template('prediction.html', predicted_crop = predicted_crop, crop_link
= crop_link , crop_img = crop_img )


    elif outcome == 12:
        predicted_crop = 'Watermelon'
        crop_link = 'https://en.wikipedia.org/wiki/Watermelon'
        crop_img = '../static/styles/images/watermelon.jpg'
```

```
        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 13:
        predicted_crop = 'Apple'
        crop_link = 'https://en.wikipedia.org/wiki/Apple'
        crop_img = '../static/styles/images/apple.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 14:
        predicted_crop = 'Orange'
        crop_link = 'https://en.wikipedia.org/wiki/Orange'
        crop_img = '../static/styles/images/orange.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 15:
        predicted_crop = 'Papaya'
        crop_link = 'https://en.wikipedia.org/wiki/Papaya'


crop_img = '../static/styles/images/papaya.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 16:
        predicted_crop = 'Coconut'
        crop_link = 'https://en.wikipedia.org/wiki/Coconut'
        crop_img = '../static/styles/images/coconut.jpg'
```

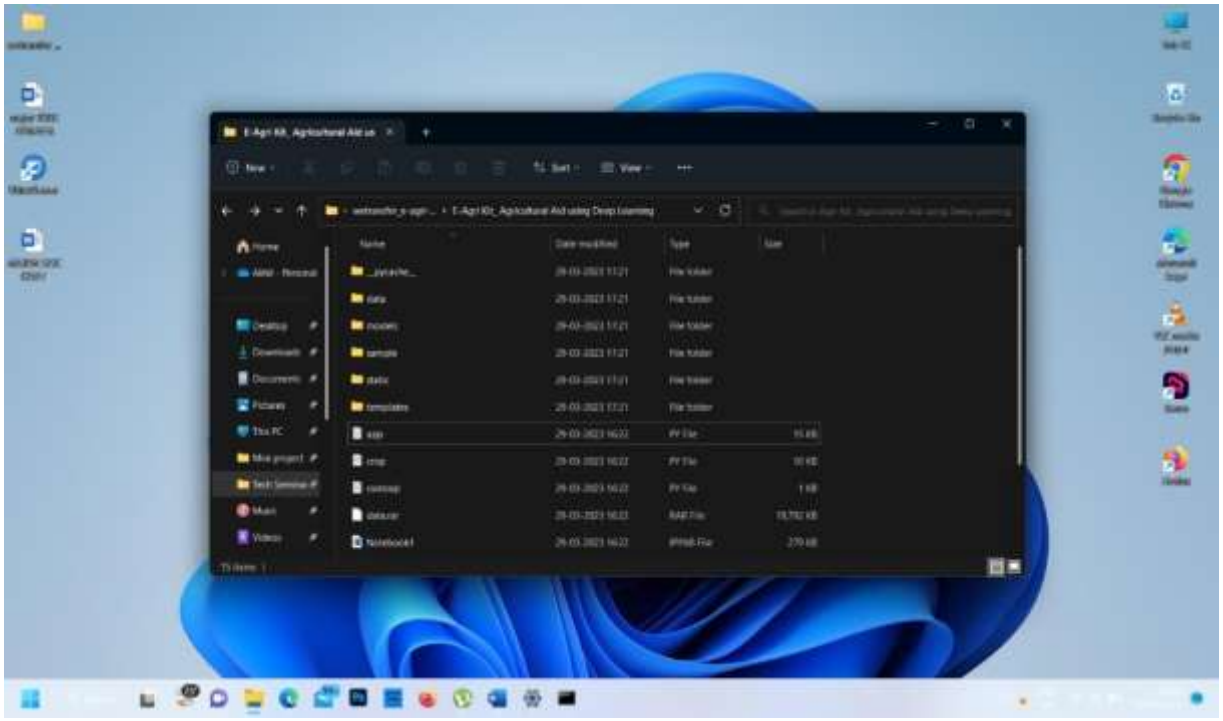
```
        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 17:
        predicted_crop = 'Cotton'
        crop_link = 'https://en.wikipedia.org/wiki/Cotton'
        crop_img = '../static/styles/images/cotton.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    elif outcome == 18:
        predicted_crop = 'Jute'
        crop_link = 'https://en.wikipedia.org/wiki/Jute'
        crop_img = '../static/styles/images/jute.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )




 elif outcome == 19:
        predicted_crop = 'Coffee'
        crop_link = 'https://en.wikipedia.org/wiki/Coffee'
        crop_img = '../static/styles/images/cofee.jpg'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


    else:
        predicted_crop = 'Tea'
```

```
        crop_link = 'https://en.wikipedia.org/wiki/Tea'
        crop_img = '../static/styles/images/'


        return render_template('prediction.html', predicted_crop = predicted_crop,
crop_link = crop_link , crop_img = crop_img )


if__name__== '_main_':
        app.run(debug=True)
```

5.RESULTS

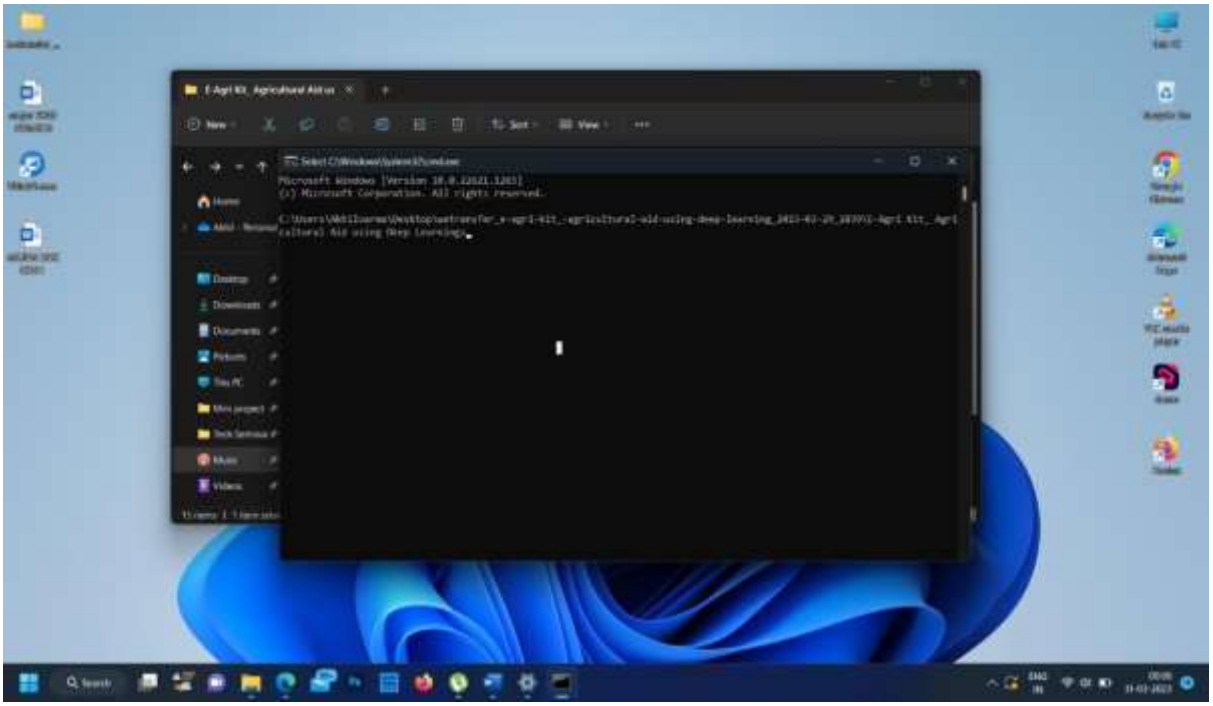
5. RESULTS

5.1 : Opening the Application



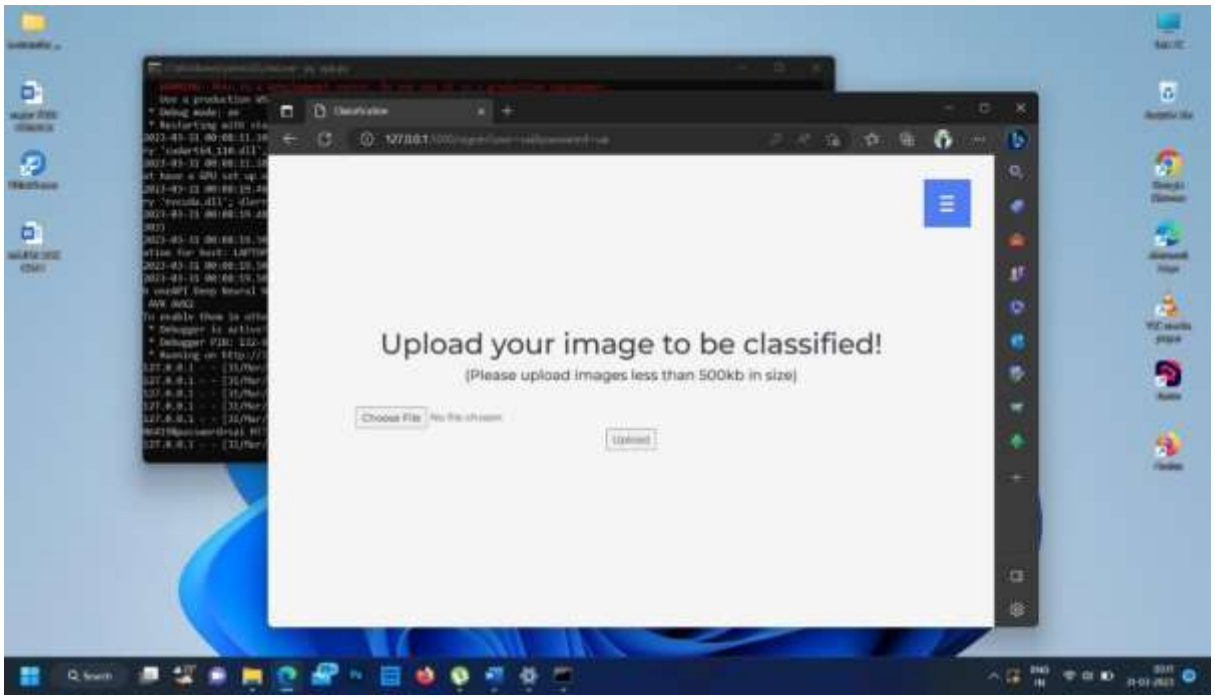
Screenshot 5.1: Opening the Application

5.2 : Running the Application



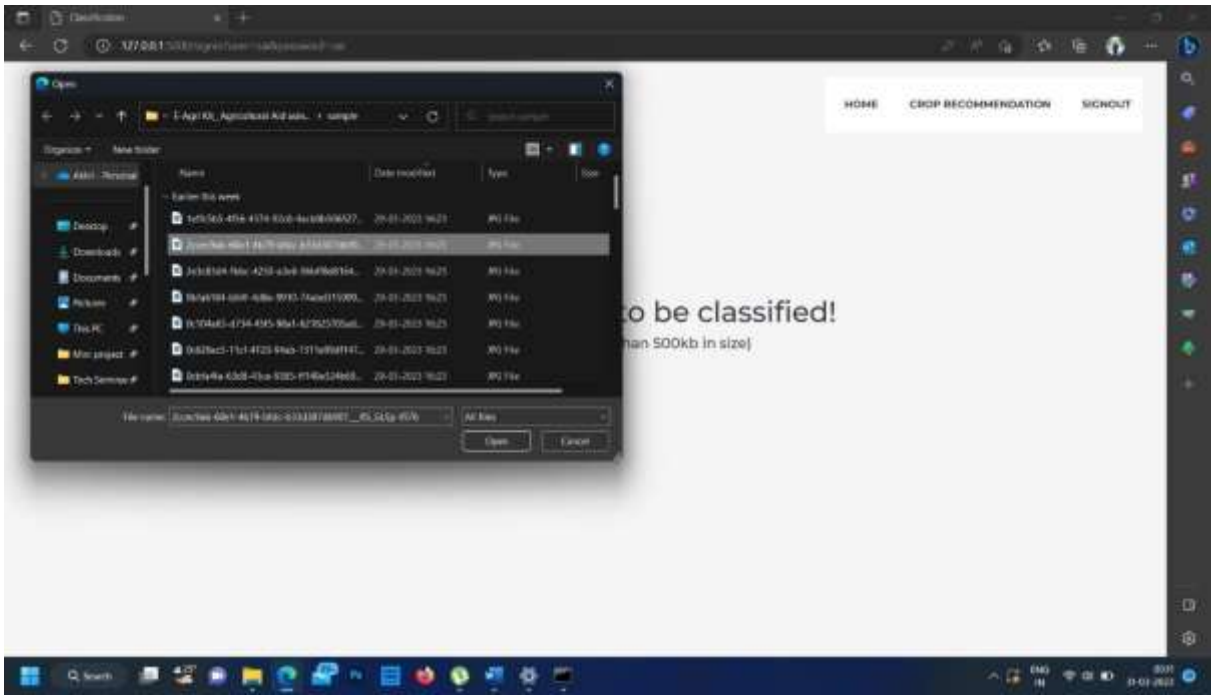
Screenshot 5.: Running the Application

5.3 : User Interface of the Application



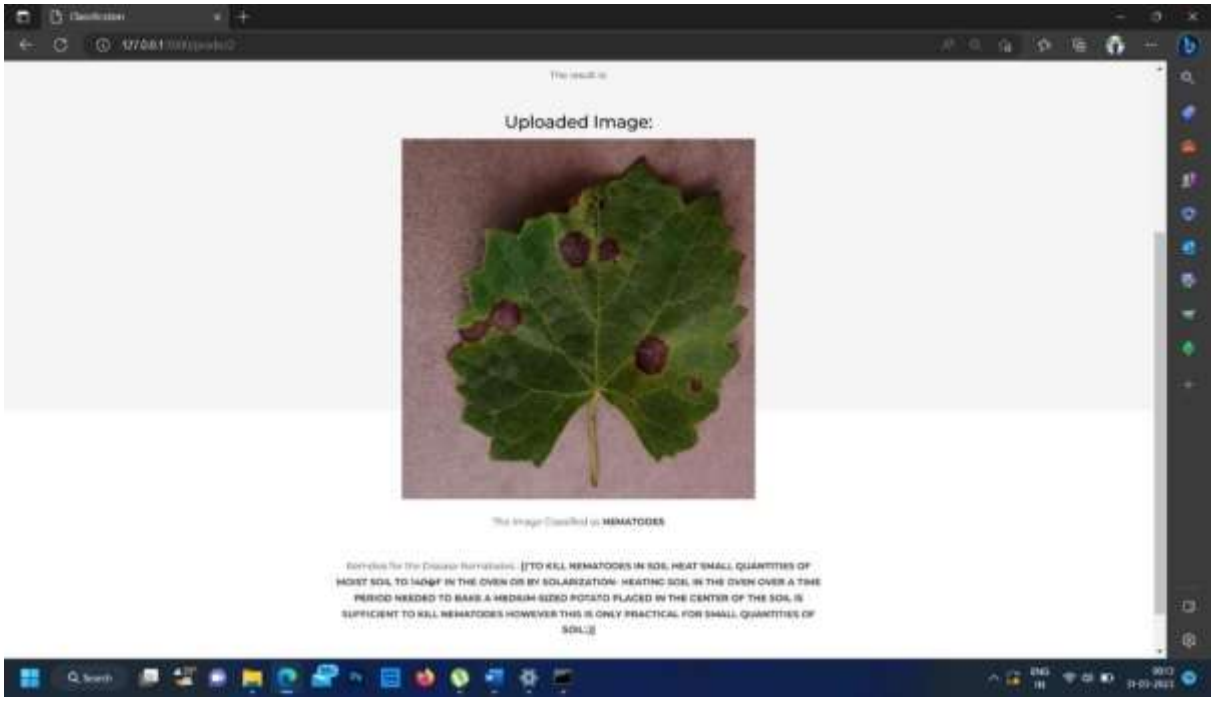
Screenshot 5.3: User Interface of the Application

5.4: Uploading Leaf Samples into the Application



Screenshot 5.4: Uploading Leaf Samples

5.5 : Detection of Disease of the Leaf Sample



Screenshot 5.5: Final Result

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organizationand preparation of functional tests is focused on requirements, key functions, orspecial test cases.

6.3TEST CASES

6.3.1 CLASSIFICATION

Test Case Id	Test Case Name	Test Case Description	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Start the Application	Host the application and test if it starts making sure the required software is available	If it doesn't Start	We cannot run the Application .	The application hosts success.	High	High
02	Home Page	Check the deployment environment for properly loading the application.	If it doesn’t load.	We cannot access the application	The application is running successfully .	High	High
03	User Mode	Verify the working of the application in freestyle	If it doesn’t Respond	We cannot use the Freestyle mode.	The application displays the Freestyle Page	High	High

		mode					
04	Data Input	Verify if the application takes input and updates	If it fails to take the input or store in The Database	We cannot proceed further	The application updates the input to application	High	High

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

During our analysis, we have understood the need for efficient plant disease identification & classification algorithms and prevention methods. Due to a large number of crops and diseases available, it is crucial that the detection system should be able to adapt to the changing variables and trends. Hence, Machine learning and Deep learning approaches were employed for this project which ensures that the code trains itself against as many possible numbers of different crops and diseases as possible The paper consists of an android application covering plant disease detection and other functionalities such as language translation, weather forecasting and fertilizer calculator. With this application we aim to provide aid in the unprecedented agricultural activities and ensure a healthy plant. We have also tested our application on cotton dataset and performed real time analysis on a diseased tomato crop to ensure our model does not overfit and performs well in a live environment. In future, we aim to expand our dataset to include more varied types of crops and disease so that the algorithm can adapt better to real time conditions and provide wide coverage.

7.2 FUTURE SCOPE

In future, we aim to expand our dataset to include more varied types of crops and disease so that the algorithm can adapt better to real time conditions and provide wide coverage.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] CropLife International (May 2015). India's farmers fighting pests. Retrieved from: https://croplife.org/news/keeping-indias-pests-in-line/

[2] Economic Times (Sept 2018). India sets record farm output target for 2018-19.

[3] Sharada P. Mohanty David P. Hughes and Marcel Salathé."Using Deep Learning for Image-Based Plant Disease Detection."Front. Plant Sci., 22 September 2016

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers (SIGGRAPH '04). Association for Computing Machinery, New York, NY, USA, 309–314.

[5] R. Chapaneri, M. Desai, A. Goyal, S. Ghose and S. Das, "Plant Disease Detection: A Comprehensive Survey," 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA),Mumbai, India, 2020, pp. 220-225, doi: 10.1109/CSCITA47329.2020.9137779.

8.2 GITHUB LINK

https://github.com/Rudhramsh/E-AGRI-KIT-AGRICULTURAL-AID-USING-DEEP-LEARNING

9.PAPER PUBLICATION

E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING

[1] Pokkili Sampath Kumar, [2] Muthe Akhil Kumar, [3] Reddy Rudhramsh Reddy,[4]Guntoju Kalpana Devi

[1,2,3] *B.Tech Student, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad,Telangana,India.sampathpokkili@gmail.com,akhilvarmamuthe@gmail.com,reddyrudhramsh@gmail.com*

[4] *Assistant Professor, Department of Computer Science and Engineering, CMR Techinical Campus, Medchal, Hyderabad, Telangana, India. gkalpana15@gmail.com*

Abstract- *This project presents an agricultural aid application, developed and designed, to help farmers by utilizing Image Processing, Machine Learning and Deep Learning concepts. Our application provides features such as early detection of plant disease, implemented using various approaches. After evaluation, results showed that Convolutional Neural Network was per forming better for plant disease detection with an high accuracy. It further helps the farmer to forecast the weather to decide the right time for agricultural activities like harvesting and plucking. To avoid reoccurrence of disease due to loss in soil minerals, a crop-specific fertilizer calculator is incorporated which can calculate the amount of urea, diammonium phosphate and muriate of potash required for a given area. This project showcases an agricultural aid app that was built and designed to assist farmers by employing Image Processing, Machine Learning, and Deep Learning. Features like early detection of plant disease are available in our application and are implemented in a number of ways. It was determined that Convolutional Neural Network was superior for detecting plant diseases with a high degree of accuracy. The farmer can use the weather forecast to plan out agricultural tasks like harvesting and plucking at the optimal time. A crop-specific fertilizer calculator is in the works to determine how much urea, diammonium phosphate, and muriate of potash should be applied to a given area to prevent the recurrence of disease caused by depleted soil minerals.*

I. INTRODUCTION

According to a study by the Associated Chambers of Commerce and Industry of India, annual crops losses due to pests and diseases amount to Rs.50,000 crore ($500 billion), which is tantamount to a country where at least 200 million go to bed hungry every night. Agriculture being a vital sector has a majority of the rural population in developing countries relying on it. The sector is faced by major challenges like unprecedented pest attack and unforeseen weather conditions affecting their produce leading to major loss of food and effort. Technology plays a vital role in uplifting the livelihoods of the rural populace which can be done by using a simple agro-android application system.Plant diseases can affect vast produce of crops posing a major menace to food security as well as leading to major losses to farmers. An extensive review of existing research was conducted by us on this domain and in an effort to help farmers overcome this problem, we have designed an android application, Agricultural Aid which utilizes machine learning to provide plant disease detection. This detection is combined with an android application which provides features like weather forecast of up to 7 days, fertilizer calculator and language translation in up to 4 languages which has been implemented and integrated using Android Studio and its APIs. For disease classification, we followed two approaches: Image Processing with Machine Learning and Deep Learning models.

The first approach i.e. Image Processing approach usually includes multi-step preprocessing techniques such as: Filtering, color space conversion, thresholding and finally, contouring to mark out the infected region. These methods can be used with Machine Learning concepts to provide classification of infected regions. However, the accuracy for such methods isn't very high. As an alternative to these steps, "GrabCut" Algorithm can also be used which is an optimized method of foreground extraction to eliminate background noises using minimal user interaction [4]. It has better accuracy in terms of background elimination and can be used for better classification however for the time being this method wasn't used in the application but can be incorporated in the future to improve accuracy. For the second approach i.e. Deep Learning approach, a deep neural architecture is used to train and test on leaf image databases to classify the disease. The paper provides a comparison of results obtained after applying Deep Learning Models such as CNN, ResNet-152 and Inception v3. In our agriculture aid,CNN Model is used to train and form an automated plant disease system based on images of leaves of both healthy and diseased plants.

II. EXISTING SYSTEM

Image Processing is a popular first step in the process of plant disease detection and often includes multi-step processes to achieve processed, ROI centric input images for further classification. In this approach, various Image Processing techniques were used on the input image to get a final output image which would mark the infected area and also calculate the percentage of area infected in the leaf. The major advantage observed in this approach was that this approach eliminated the need to extract the leaf and place it on a black background during image capture for the algorithm to work. In real time scenario, the infected leaf would be present amongst a cluster of mixed crops and this approach was able to segregate the infected leaf from the healthy ones in an input image.

III. PROPOSED SYSTEM

In the Deep Learning Approach, we decided to take a subset of the Plant Village dataset along with the cotton dataset to train and test the CNN model. The input image is fed into this model, which initially took a portion of the Plant Village and Cotton Dataset, with a training -validation split of 70-30, therefore getting 4200 images for training and 1800 images for validation. A CNN (Convolutional Neural Network) is a deep learning model that takes inputs which are assigned weights depending on various features. CNN is a widely used neural network for image-based datasets.

Our CNN model consists of 4 main convolutional layers with 32, 64, 128, 128 filters consecutively, each followed by a ReLU activation function, max pooling and dropout layer. This set of convolutional layers is followed by a flatten and then a dense layer which is finally followed by the soft max activation function that tells us which class has the maximum probability.

IV. FEASIBILITY STUDY

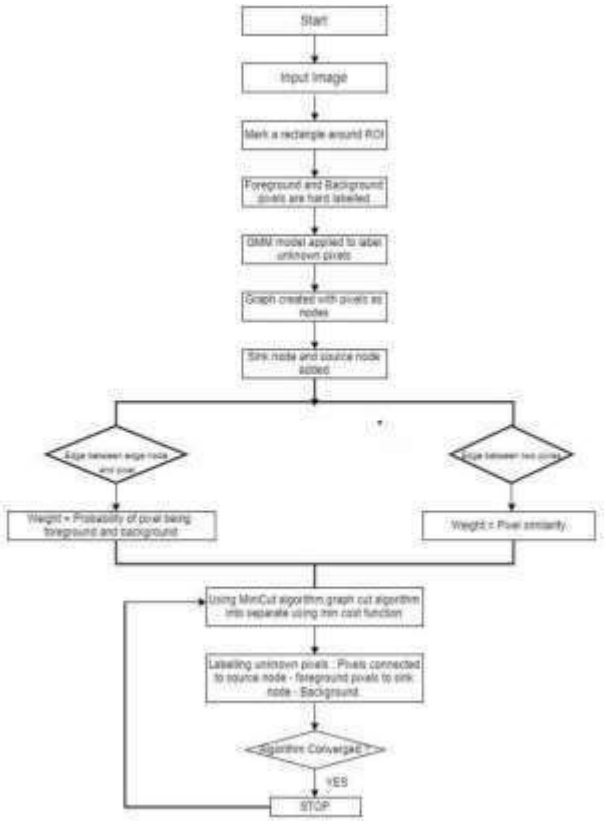
The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

*** ECONOMICAL FEASIBILITY :** This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

*** TECHNICAL FEASIBILTY :** This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

*** SOCIAL FEASIBILITY :** The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

V. ARCHITECTURE



After selecting the image for extraction, the user is required to mark out a rectangle selecting the region/object of interest (ROI) where everything outside the rectangle is taken as background. The computer labels the pixels as

"foreground" and "background" pixels which is also known as hard labelling. After labelling, a Gaussian Mixture Model (GMM) is applied on the foreground and background. GMM learns and tries to label unknown pixels inside the rectangle as either probably foreground or background according to hard-labelled pixels and color statistic which is similar to clustering. Using this pixel distribution, a graph is created where pixels are denoted as nodes. Additionally, 2 nodes are added, namely: Source node and Sink node such that every foreground pixel is connected to source node and every background pixel is connected to sink node. The weight of the edges connecting pixels to source node is calculated as the probability of the pixel being a part of the foreground or the background. The weight of the edges connecting two pixels is defined by pixel similarity such that if there is a small difference in pixel color the weight will be high. Using a min-cut algorithm on the graph, the graph is cut into separate source node and sink node using minimum cost function where the cost function is sum of all the edges that are cut. After the cut, nodes (pixels) connected to source node are labelled as foreground and nodes(pixels) connected to sink node becomes background.

VI. MODULES

1.Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.Tensor Flow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

2. Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various
features including these important ones:

- ➢ A powerful N-dimensional array object
- ➢ Sophisticated (broadcasting) functions
- ➢ Tools for integrating C/C++ and Fortran code
- ➢ Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

3. Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem.

4. Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc.,with just a few lines of code. For examples, see the sample plots and thumbnail gallery.For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

5. Scikit – learn

Scikit-learn provide a standard Python interface for a variety of supervised and unsupervised learning techniques. It is licensed under a BSD-style license that allows for both academic and commercial use, and is available on a variety of Linux versions.

VII. CONCLUSION

During our analysis, we have understood the need for efficient plant disease identification & classification algorithms and prevention methods. Due to a large number of crops and diseases available, it is crucial that the detection system should be able to adapt to the changing variables and trends. Hence, Machine learning and Deep learning approaches were employed for this project which ensures that the code trains itself against as many possible numbers of different crops and diseases as possible The paper consists of an android application covering plant disease detection and other functionalities such as language translation, weather forecasting and fertilizer calculator. With this application we aim to provide aid in the unprecedented agricultural activities and ensure a healthy plant. We have also tested our application on cotton dataset and performed real time analysis on a diseased tomato crop to ensure our model does not overfit and performs well in a live environment. In future, we aim to expand our dataset to include more varied types of crops and disease so that the algorithm can adapt better to real time conditions and provide wide coverage.

ACKNOWLEDGMENT

We thank CMR Technical Campus for supporting this paper titled "E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING", which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head Of the Department, Department Of Computer

Science and Engineering, Guide and Teaching and Non-Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work.

REFRENCES

1. CropLife International (May 2015). India's farmers fighting pests. Retrieved from:https://croplife.org/news/keeping-indias-pests-in-line/
2. Economic Times (Sept 2018). India sets record farm output target for 2018-19. Retrieved from: https://economictimes.indiatimes.com/news/economy/agriculture/indiasets-record-farm-output-target-for-2018-19/articleshow/65858058.cms
3. Sharada P. Mohanty David P. Hughes and Marcel Salathé."Using Deep Learning for Image-Based Plant Disease Detection."Front. Plant Sci., 22 September 2016.
4. Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers (SIGGRAPH '04). Association for Computing Machinery, New York, NY, USA, 309–314.
5. R. Chapaneri, M. Desai, A. Goyal, S. Ghose and S. Das, "Plant Disease Detection: A Comprehensive Survey," 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA),Mumbai, India, 2020, pp. 220-225, doi: 10.1109/CSCITA47329.2020.9137779.
6. Raghavendra, B. K. (2019, March). Diseases Detection of Various Plant Leaf Using Image Processing Techniques: A Review. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), (pp. 313-316). IEEE.

10.CERTIFICATES

10.CERTIFICATES

Journal of Interdisciplinary Cycle Research

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0022-1945 / web : http://jicrjournal.com / e-mail: submitjicrjournal@gmail.com

Certificate of Publication

This is to certify that the paper entitled Certificate Id: JICR/7024

"E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING"

Authored by :

Guntoju Kalpana Devi, Assistant Professor

From

CMR TECHNICAL CAMPUS

Has been published in

JICR JOURNAL, VOLUME XV, ISSUE IV, APRIL-2023

Dr. R. Rezwana Begum, Ph.D Editor-In-Chief

JICR JOURNAL

6.2

ISO 7021-2008

Journal of Interdisciplinary Cycle Research

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0022-1945 / web : http://jicrjournal.com / e-mail: submitjicrjournal@gmail.com

Certificate of Publication

This is to certify that the paper entitled Certificate Id: JICR/7024

"E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING"

Authored by :

Pokkili Sampath Kumar

From

CMR TECHNICAL CAMPUS

Has been published in

JICR JOURNAL, VOLUME XV, ISSUE IV, APRIL-2023

Dr. R. Rezwana Begum, Ph.D Editor-In-Chief

JICR JOURNAL

6.2

ISO 7021-2008

Journal of Interdisciplinary Cycle Research

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0022-1945 / web : http://jicrjournal.com / e-mail: submitjicrjournal@gmail.com

Certificate of Publication

This is to certify that the paper entitled Certificate Id: JICR/7024

"E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING"

Authored by :

muthe Akhil Kumar

From

CMR TECHNICAL CAMPUS

Has been published in

JICR JOURNAL, VOLUME XV, ISSUE IV, APRIL-2023

Dr. R. Rezwana Begum, Ph.D Editor-In-Chief

JICR JOURNAL

6.2

ISO 7021-2008

Journal of Interdisciplinary Cycle Research

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0022-1945 / web : http://jicrjournal.com / e-mail: submitjicrjournal@gmail.com

Certificate of Publication

This is to certify that the paper entitled Certificate Id: JICR/7024

"E-AGRI KIT AGRICULTURAL AID USING DEEP LEARNING"

Authored by :

Reddy Rudhramsh Reddy

From

CMR TECHNICAL CAMPUS

Has been published in

JICR JOURNAL, VOLUME XV, ISSUE IV, APRIL-2023

Dr. R. Rezwana Begum, Ph.D Editor-In-Chief

JICR JOURNAL

6.2

ISO 7021-2008

http://jicrjournal.com