A Project report on

AGRICULTURE CROP IMAGE CLASSIFICATION

A Dissertation submitted to JNTUH, Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Artificial Intelligence and Machine Learning

Submitted by

K. LAXMI SANDEEP
(21H51A7332)

L. SATHWIK
(21H51A7333)

P. NAVEEN KUMAR
(21H51A7342)

Under the esteemed guidance of

Dr. J. SASI BHANU
(Professor)

Department of Artificial Intelligence and Machine Learning

CMR COLLEGE OF ENGINEERING& TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2024-2025

.

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the Major Project report entitled **"Agriculture Crop Image Classification"** being submitted by K.Laxmi Sandeep (21H51A7332), L.Sathwik (21H51A7333), P.Naveen Kumar (21H51A7342) in partial fulfillment for the award of **Bachelor of Technology in Artificial Intelligence and Machine Learning** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr. J. Sasi Bhanu
Professor
Dept. Of AIML

Dr. S. Kirubakaran
Professor&HOD
Dept. Of AIML

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success

We are grateful to **Dr. J. Sasi Bhanu, Professor** , Department of Artificial Intelligence and Machine Learning for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. S. Kirubakaran,** Head of the Department of Artificial Intelligence and Machine Learning, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We would like to thank **Dr. P. Ravi Kumar**, Dean F&S, CMR College of Engineering and Technology, for his insight and expertise have been instrumental in shaping the direction and execution of this project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We extend our heartfelt gratitude to **Dr.Seshu Kumar Avadhanam**, Principal, CMR College of Engineering and Technology, for his unwavering support and guidance in the successful completion of our project and his encouragement has been invaluable throughout this endeavor.

We are highly indebted to **Major Dr. V A Narayana,** Director, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and **Shri Ch Abhinav Reddy**, CEO, CMR Group of Institutions for their continuous care and support

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

K. Laxmi Sandeep	21H51A7332
L. Sathwik	21H51A7333
P. Naveen Kumar	21H51A7342

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Agriculture crop image classification is a critical task in precision farming, enabling improved crop management, disease monitoring, and resource optimization. Traditional methods rely heavily on manual observations, which are time-consuming and prone to errors. This project leverages deep learning techniques to automate and enhance the accuracy of crop image classification. Using Convolutional Neural Networks (CNNs), the system analyzes and identifies crop types from images with high precision. A diverse dataset of crop images was utilized, and data augmentation techniques were employed to ensure robustness against varying environmental conditions. Transfer learning further optimized model performance, reducing training time and computational requirements. The results demonstrate the effectiveness of the proposed approach, offering a scalable and automated solution for crop identification. This system has significant potential to aid farmers, agronomists, and policymakers by providing real-time insights, fostering sustainable farming practices, and advancing the adoption of artificial intelligence in agriculture.

The proposed system demonstrates scalability, accuracy, and real-time applicability, making it a valuable tool for farmers, agronomists, and policymakers. It can facilitate better decision-making in crop monitoring, disease detection, and resource allocation. The project highlights the potential of integrating artificial intelligence into agriculture, paving the way for advancements in precision farming and sustsainable practices. Future enhancements include expanding the dataset to cover additional crop types, integrating IoT-based real-time monitoring systems, and exploring multi-modal approaches to boost classification accuracy further.

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Agriculture plays a vital role in sustaining economies and societies worldwide. One of the significant challenges faced by farmers and agricultural researchers is the identification and classification of crops based on their visual characteristics. Manual classification methods are labour-intensive, time-consuming, and prone to errors, especially in large-scale agricultural practices. Moreover, factors such as crop similarity, variations in lighting, and differences in growth stages further complicate the task. With the advent of deep learning, there is a promising opportunity to automate crop classification using image-based methods. However, implementing such systems in agricultural settings requires overcoming challenges like diverse environmental conditions, limited labelled datasets, and computational constraints. Addressing these issues is crucial to ensure reliable, scalable, and cost-effective solutions for modern agriculture.

1.2 Research Objective

The specific objectives are as follows:

1. Design and implement a **Convolutional Neural Network (CNN)** model capable of accurately classifying crops based on image data.
2. Enhance the dataset through **image preprocessing** techniques such as augmentation and normalization to improve the model's robustness.
3. Evaluate the model's performance using metrics like **accuracy**, **precision**, **recall**, and **F1-score** to ensure comprehensive validation.
4. Provide a user-friendly framework that can assist stakeholders in agriculture, such as farmers, researchers, and agricultural agencies, in making informed decisions.
5. Explore potential extensions, such as incorporating **multi-spectral imagery** or expanding the model to detect diseases and pest infestations.

1.3 Project Scope and Limitations

Scope:

- **Dataset**: The project focuses on a dataset of 20 different crops with 30 images each, representing a variety of commonly grown crops.
- **Deep Learning Techniques**: Utilizes state-of-the-art CNN architectures for feature extraction and classification.
- **Applications**: The developed model can be deployed for tasks such as crop monitoring, precision agriculture, and agricultural planning.
- **Scalability**: The system provides a foundation for expanding to broader agricultural datasets, integrating satellite imagery, and adapting to different environmental conditions.

Limitations:

- **Dataset Diversity**: The dataset may not fully capture variations such as lighting, seasonal changes, or geographic differences.
- **Hardware Requirements**: Training and deploying deep learning models require significant computational resources, potentially limiting accessibility in resource-constrained settings.
- **Scalability:** While the approach is tailored for image encryption, its adaptability to other data types (e.g., video, text) requires further exploration.
- **Model Generalization**: The model's performance might degrade when applied to unseen datasets or crop types not included in the training data.
- **Environmental Factors**: Real-world conditions, such as overlapping plants or poor image quality, might affect the accuracy of the classification.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. Existing Method 1: Manual Crop Classification by Experts

2.1.1. Introduction

Manual crop classification relies on human expertise to identify and classify crops based on physical features such as leaf shape, size, color, and growth patterns. This method has been widely used in traditional agricultural practices and research due to its reliance on observable traits. It is often performed by agricultural scientists or field experts who gather data by visiting farms and observing crops directly.

2.1.2. Merits, Demerits, and Challenges

- **Merits**:
  - No requirement for advanced technological infrastructure.
  - Provides high accuracy in localized regions due to experts' familiarity with specific crops.
  - Useful for regions lacking access to digital tools.
- **Demerits**:
  - **Labor-Intensive:** Requires significant manual effort and time for large-scale farming.
  - **Subjective Variability:** Results may vary between experts due to differences in experience and skill levels.
  - **Scalability Issues:** Inefficient for large-scale operations involving diverse crops.

- **Challenges:**
    - The lack of sufficient experts in rural or remote regions.
    - Difficulty in maintaining consistency in classifications across different geographic regions.
    - High costs and inefficiencies in large-scale farming scenarios.

2.1.3. Implementation of Existing Method 1

Manual crop classification is typically implemented through the following steps:

[1]. **Data Collection:** Experts visit farms to gather observations on crop characteristics.

[2]. **Classification:** Based on visual inspection and predefined criteria, experts assign crops to categories.

[3]. **Reporting:** The results are documented and analyzed for decision-making purposes.

Fig 2.1.3: Manual Crop Identification

2.2. Existing Method 2: Rule-Based Image Classification

2.2.1. Introduction

The 3D Rule-based classification systems rely on predefined criteria, such as pixel intensity and color thresholds, to identify crop types from images. These systems are straightforward and do not require complex algorithms or high computational power.

2.2.2. Merits, Demerits, and Challenges

- **Merits**:
  - Simple and easy to implement.
  - Requires minimal computational resources.
  - Effective for small datasets with well-defined rules.
- **Demerits**:
  - Limited adaptability to diverse datasets.
  - Performs poorly with noisy or low-quality images.
  - Cannot handle complex patterns or overlapping crops.
- **Challenges:**
  - Difficulty in updating rules to accommodate new crop types.
  - Limited scalability and generalization for large-scale applications.

2.2.3. Implementation of Existing Method 2

- **Define Rules:** Establish specific rules based on crop characteristics like color and texture.
- **Apply Rules:** Analyze image data using the defined rules.
- **Output:** Classify crops based on the criteria.

2.3 Existing Method 3: Machine Learning-Based Classification

2.3.1 Introduction

Machine learning methods use algorithms like Decision Trees, Random Forest, or Support Vector Machines (SVM) to classify crops based on features extracted from images. These methods rely on labelled datasets to train models and predict crop types.

2.3.2 Merits, Demerits, and Challenges

Merits:

- Higher accuracy compared to manual and rule-based methods.
- Can handle moderately large datasets and complex patterns.
- Adaptable for various crops and regions.

Demerits:

- Requires labelled data for training, which can be time-consuming to prepare.
- Performance depends on the quality of feature selection.
- Computationally expensive for large datasets.

Challenges:

- Difficulty in distinguishing crops with similar visual features.
- Requires expertise in machine learning techniques and tools.

2.3.3 Implementation of Existing Method 3

1. **Data Preparation:** Gather and preprocess crop images to extract features.
2. **Model Training:** Train machine learning models using labeled datasets.
3. **Prediction:** Use the trained model to classify new crop images.

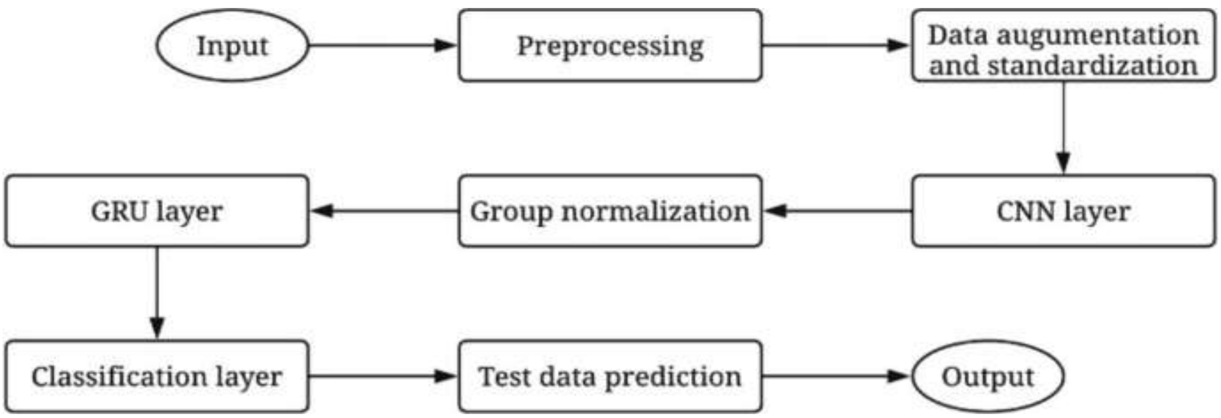


Fig 2.3.3: Work flow of crop classification

CHAPTER 3
PROPOSED SYSTEM

3.1 Research Objective of Proposed Model

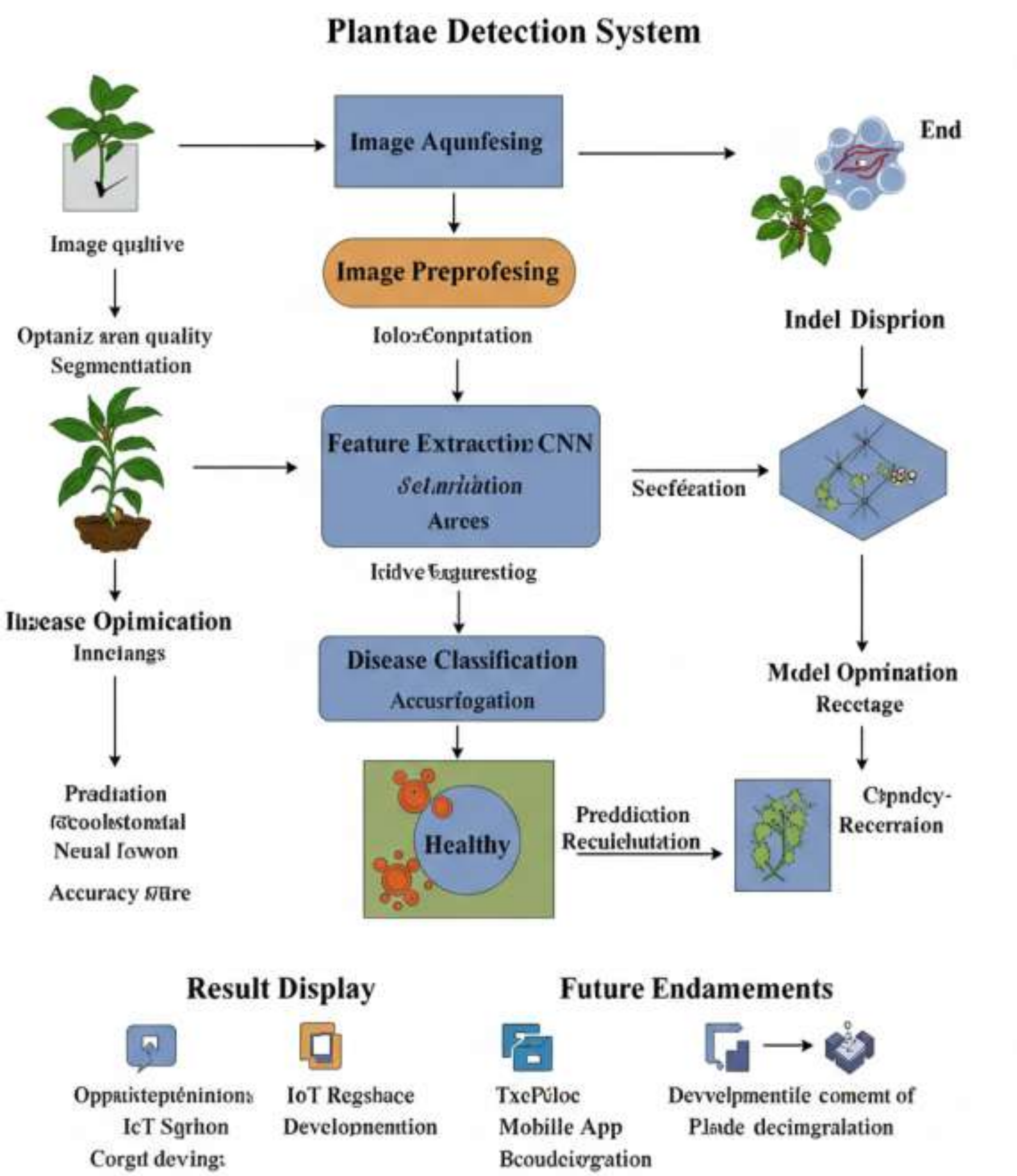
1. **Improve Classification Accuracy Across Crop Types**: To enhance model performance by training on a diverse dataset representing multiple crop species, growth stages, and environmental conditions, thereby ensuring the model's reliability in real-world applications.
2. **Develop an Effective Image Classification Model**: To design and implement a machine learning model, specifically leveraging convolutional neural networks (CNNs), capable of accurately classifying various agricultural crop types from image data.
3. **Evaluate Model Generalization and Adaptability**: To test the model's adaptability across diverse agricultural contexts, ensuring that it remains effective across different geographic locations, crop varieties, and climatic conditions.

3.2 Algorithms Used for Proposed Model

1. **The Residual Network (ResNet) :** The Residual Network (ResNet) is a deep neural network architecture developed to tackle the vanishing gradient issues in extremely deep neural networks. ResNet incorporates residual learning via skip connections so that layers learn residual mapping rather than direct feature transformations. This drastically enhances the flow of gradients, enabling very deep networks to be easily trained while keeping accuracy intact. ResNet layers multiple batch-normalization convolutions together and uses ReLU activations to enable effective plant disease image feature extraction. Compared with conventional CNN, ResNet provides superior classification results due to the conservation of low- and high-level features across the depth of layers.
2. **Convolutional Neural Network (CNN)**: In this project, a Convolutional Neural Network (CNN) was implemented to classify agricultural crop diseases based on image data. CNN is a deep learning algorithm that is highly effective for image recognition tasks due to its ability to automatically extract relevant features from image

3.3 Designing

3.3.1 UML Diagram



3.4. Stepwise Implementation and Code

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from tkinter import filedialog
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.layers.core import Dense, Activation, Dropout, Flatten
from sklearn.metrics import accuracy_score
import webbrowser
import os
import cv2
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
import pickle
from keras.models import model_from_json

main = Tk()
main.title("Agriculture Crop Image Classification")
main.geometry("1300x1200")

global filename
global X, Y
global model
global accuracy

plants = ['Pepper_bell_Bacterial_spot', 'Pepperbell_healthy', 'Potato_Early_blight', 'Potato__healthy',
        'Potato__Late_blight', 'TomatoTarget_Spot', 'Tomato_Tomato_mosaic_virus',
        'Tomato_Tomato_YellowLeaf_Curl_Virus', 'Tomato_Bacterial_spot', 'Tomato_Early_blight',
'Tomato_healthy',
        'Tomato_Late_blight', 'Tomato_Leaf_Mold', 'Tomato_Septoria_leaf_spot',
        'Tomato_Spider_mites_Two_spotted_spider_mite',
        'Brinjal_Epilachna_Beetle', 'Brinjal_Flea_Beetle', 'Brinjal_Healthy', 'Brinjal_Jassid',
'Brinjal_Mite',
        'Brinjal_Mite_and_Epilachna_Beetle', 'Brinjal_Nitrogen_and_Potassium_Deficiency',
'Brinjal_Nitrogen_Deficiency',
        'Brinjal_Potassium_Deficiency']

disease_info = {
   'Pepper_bell__Bacterial_spot': {
      'Precautions': ['Plant disease-resistant pepper varieties', 'Avoid overhead watering'],
      'Pesticides': 'Double Nickel 55 LC ,WDG,EcoSwing Botanical Fungicide,GreenFurrow BacStop.'
   },
   'Pepper_bell__healthy': {
```

```
        'Precautions': ['Practice good pepper plant hygiene', 'Control pests'],
        'Pesticides': 'No Pesticides needed, as the plant is healthy.'
    },
    'Potato___Early_blight': {
        'Precautions': ['Plant disease-resistant potato varieties', 'Apply fungicides'],
        'Pesticides': 'Chlorothalonil, Famoxadone/Cymoxanil (Tanos),Mancozeb.'
    },
    'Potato___healthy': {
        'Precautions': ['Practice good potato plant hygiene', 'Control pests'],
        'Pesticides': 'No Pesticides needed, as the plant is healthy.'
    },
    'Potato___Late_blight': {
        'Precautions': ['Plant disease-resistant potato varieties', 'Apply fungicides'],
        'Pesticides': 'Fenamidone (Reason 500SC),Azoxystrobin (Quadris), Boscalid (Endura).'
    },
    'Tomato__Target_Spot': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Prune affected leaves'],
        'Pesticides': 'chlorothalonil, mancozeb, and copper oxychloride.'
    },
    'Tomato__Tomato_mosaic_virus': {
        'Precautions': ['Control aphid vectors', 'Remove infected plants'],
        'Pesticides': 'There is no Pesticides for this viral disease; remove and destroy infected plants.'
    },
    'Tomato_Tomato_YellowLeaf_Curl_Virus': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Prune affected leaves'],
        'Pesticides': 'cypermethrin, deltamethrin, bifenthrin.'
    },
    'Tomato_Bacterial_spot': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Prune infected branches'],
        'Pesticides': 'acibenzolar-S-methyl (Actigard®), Serenade Opti®, and Sporan EC2.'
    },
    'Tomato_Early_blight': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Apply fungicides'],
        'Pesticides': 'copper products mixed with mancozeb, as well as biopesticides like Serenade Opti®.'
    },
    'Tomato_healthy': {
        'Precautions': ['Practice good tomato plant hygiene', 'Control pests'],
        'Pesticides': 'No Pesticides needed, as the plant is healthy.'
    },
    'Tomato_Late_blight': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Apply fungicides'],
        'Pesticides': 'fungicides like copper (Champ), chlorothalonil (Bravo).'
    },
    'Tomato_Leaf_Mold': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Prune affected leaves'],
        'Pesticides': 'chemical fungicides based on sulfur, as well as natural fungicides like neem oil,
rosemary oil.'
    },
    'Tomato_Septoria_leaf_spot': {
        'Precautions': ['Plant disease-resistant tomato varieties', 'Prune infected leaves'],
        'Pesticides': 'maneb, mancozeb, and benomyl.'
    },
    'Tomato_Spider_mites_Two_spotted_spider_mite': {
        'Precautions': ['Monitor and control spider mite populations', 'Maintain good plant health'],
```

```
        'Pesticides': 'Apply insecticidal soap or neem oil,insecticidal soaps, horticultural oils.'
    }
}

def uploadDataset():
    global X, Y
    global filename
    text.delete('1.0', END)
    filename = filedialog.askdirectory(initialdir=".")
    text.insert(END, 'dataset loaded\n')

def imageProcessing():
    text.delete('1.0', END)
    global X, Y
    X = np.load("model/myimg_data.txt.npy")
    Y = np.load("model/myimg_label.txt.npy")
    Y = to_categorical(Y)
    X = np.asarray(X)
    Y = np.asarray(Y)
    X = X.astype('float32')
    X = X / 255
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    text.insert(END, 'image processing completed\n')
    img = X[20].reshape(64, 64, 3)
    cv2.imshow('ff', cv2.resize(img, (250, 250)))
    cv2.waitKey(0)

def cnnModel():
    global model
    global accuracy
    text.delete('1.0', END)
    if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            model = model_from_json(loaded_model_json)
        json_file.close()
        model.load_weights("model/model_weights.h5")
        model._make_predict_function()
        print(model.summary())
        f = open('model/history.pckl', 'rb')
        accuracy = pickle.load(f)
        f.close()
        acc = accuracy['accuracy']
        acc = acc[9] * 100
        text.insert(END, "CNN Crop Disease Recognition Model Prediction Accuracy = " + str(acc))
    else:
        model = Sequential()
        model.add(Convolution2D(32, 3, 3, input_shape=(64, 64, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Convolution2D(32, 3, 3, activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
    model.add(Flatten())
    model.add(Dense(output_dim=256, activation='relu'))
    model.add(Dense(output_dim=25, activation='softmax'))  # Updated output_dim to 25
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    print(model.summary())
    hist = model.fit(X, Y, batch_size=16, epochs=10, validation_split=0.2, shuffle=True, verbose=2)
    model.save_weights('model/model_weights.h5')
    model_json = model.to_json()
    with open("model/model.json", "w") as json_file:
        json_file.write(model_json)
    json_file.close()
    f = open('model/history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
    f = open('model/history.pckl', 'rb')
    accuracy = pickle.load(f)
    f.close()
    acc = accuracy['accuracy']
    acc = acc[9] * 100
    text.insert(END, "CNN Crop Disease Recognition Model Prediction Accuracy = " + str(acc))

selected_disease = ""  # Declare selected_disease as a global variable

def predict():
  global model
  global selected_disease  # Access the global variable

  filename = filedialog.askopenfilename(initialdir="testImages")
  img = cv2.imread(filename)
  img = cv2.resize(img, (64, 64))
  im2arr = np.array(img)
  im2arr = im2arr.reshape(1, 64, 64, 3)
  test = np.asarray(im2arr)
  test = test.astype('float32')
  test = test / 255
  preds = model.predict(test)
  predict = np.argmax(preds)
  img = cv2.imread(filename)
  img = cv2.resize(img, (800, 400))
  cv2.putText(img, 'Crop Disease Recognize as : ' + plants[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX, 0.7,
        (0, 255, 0), 2)
  cv2.imshow('Crop Disease Recognize as : ' + plants[predict], img)
  cv2.waitKey(0)

  # Set the selected disease
  selected_disease = plants[predict]

  # Display precautions and Pesticides for the detected disease
  if selected_disease in disease_info:
    precautions = "\n".join(disease_info[selected_disease]['Precautions'])
    Pesticides = disease_info[selected_disease]['Pesticides']
    text.insert(END, f"\nDisease: {selected_disease}\n", 'disease')
    text.insert(END, f"Precautions:\n{precautions}\n", 'precaution')
```

```
        text.insert(END, f"Pesticides:\n{Pesticides}\n", 'Pesticides')
        text.tag_configure("disease", foreground="red")
        text.tag_configure("precaution", foreground="blue")
        text.tag_configure("Pesticides", foreground="green")

def open_video():
    global selected_disease  # Access the global variable

    # Dictionary mapping diseases to YouTube video links
    disease_videos = {
        'Pepper_bell__Bacterial_spot': 'https://youtu.be/1HgsMF4gd7U?si=xJPxwxKyXfHyEh5W',
        'Pepper_bell__healthy': 'https://www.youtube.com/watch?v=Mr6i4s5bSA',
        'Potato___Early_blight': 'https://youtu.be/6i5_sLY_pWc?si=tq_rzMe1YPkeCf5w',
        'Potato___healthy': 'https://www.youtube.com/watch?v=bGq3bNWwGAs',
        'Potato___Late_blight': 'https://youtu.be/PSXXoGrOyDg?si=-vpOEjSSBnuMG1oB',
        'Tomato__Target_Spot': 'https://youtu.be/jONPVKSvFW0?si=L5lULWoGYDZf_KvL',
        'Tomato__Tomato_mosaic_virus': 'https://youtu.be/n8PyKuGcURY?si=TbNT8ZwIUsiE8Ves',
        'Tomato_Tomato_YellowLeaf_Curl_Virus':
'https://youtu.be/cM5jEqd5n5A?si=NHsNhqiVRysx_uAb',
        'Tomato_Bacterial_spot': 'https://youtu.be/JT1gGgJmOBM?si=FdqcM-ngtoUDixiY',
        'Tomato_Early_blight': 'https://youtu.be/2GYD7aVBFtg?si=-lACqKF_KVUuAF2G',
        'Tomato_healthy': 'No video link available',
        'Tomato_Late_blight': 'https://youtu.be/Djd7z03iSYE?si=7NVGT-ArQnS9K-z_',
        'Tomato_Leaf_Mold': 'https://youtu.be/0lZOboTH8m4?si=2cwr43FsMfgXUxrq',
        'Tomato_Septoria_leaf_spot': 'No video link available',
        'Tomato_Spider_mites_Two_spotted_spider_mite': 'https://youtu.be/iRYvw9vRguk?si=cRLo6w3-
hbQXyVtX'
    }

    # Check if the selected disease has a corresponding video link
    if selected_disease in disease_videos:
        video_url = disease_videos[selected_disease]
        webbrowser.open_new(video_url)
    else:
        messagebox.showinfo("Video Not Available", "Sorry, video tutorial for this disease is not
available.")


def graph():
    acc = accuracy['accuracy']
    loss = accuracy['loss']
    plt.figure(figsize=(10, 6))
    plt.grid(True)
    plt.xlabel('Iterations')
    plt.ylabel('Accuracy/Loss')
    plt.plot(acc, 'ro-', color='green')
    plt.plot(loss, 'ro-', color='blue')
    plt.legend(['Accuracy', 'Loss'], loc='upper left')
    plt.title('Iteration Wise Accuracy & Loss Graph')
    plt.show()

def close():
    main.destroy()
    text.delete('1.0', END)
```

```
font = ('times', 15, 'bold')
title = Label(main, text='Agriculture Crop Image Classification')
title.config(font=font)
title.config(height=3, width=120)
title.config(bg="#87CEEB")
title.place(x=0, y=5)

font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')


uploadButton = Button(main, text="1.Upload Crop Disease Dataset", command=uploadDataset)
uploadButton.place(x=300, y=100)
uploadButton.config(font=ff)

processButton = Button(main, text="2.Image Processing & Normalization",
command=imageProcessing)
processButton.place(x=600, y=100)
processButton.config(font=ff)

modelButton = Button(main, text="3.Build Crop Disease Recognition Model", command=cnnModel)
modelButton.place(x=900, y=100)
modelButton.config(font=ff)

predictButton = Button(main, text="4.Upload Test Image & Predict Disease", command=predict)
predictButton.place(x=300, y=150)
predictButton.config(font=ff)

graphButton = Button(main, text="5.Accuracy & Loss Graph", command=graph)
graphButton.place(x=600, y=150)
graphButton.config(font=ff)

videoButton = Button(main, text="6.video", command=open_video)
videoButton.place(x=900, y=150)
videoButton.config(font=ff)

exitButton = Button(main, text="7.Exit", command=close)
exitButton.place(x=250, y=700)
exitButton.config(font=ff)


font1 = ('times', 12, 'bold')
text = Text(main, height=30, width=85)
scroll = Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=350, y=200)
text.config(font=font1)

main.config(bg='#002147')
main.mainloop()
```

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Performance metrics

Metric	Manual Classification	Rule-Based Image Classification	Machine Learning-Based Classification
Visual Quality	High visual quality due to expert analysis of images.	Medium visual quality; simple image processing methods used.	High visual quality; data can be enriched with high-resolution images.
Data Capacity	Limited data capacity, as it's manually handled.	Moderate data capacity; depends on image size and rule sets used.	Very high data capacity; models can handle massive datasets efficiently.
Computational Efficiency	Low computational efficiency; human labor-intensive.	High efficiency for small datasets; rule-based methods are fast.	Low computational efficiency; requires significant computational resources for training and processing.
Security Level	Low security; data is handled by individuals and may lack encryption.	Moderate security; rule sets can be shared, and data may not be secure.	High security; models and datasets can be encrypted and protected during training and inference
Speed	Slow; depends on the availability of experts.	Fast; rule-based systems are quick to process small datasets.	Slow; model training and inference can be time-consuming for large datasets.
Robustness	Low robustness; relies on expert knowledge, which is error-prone.	Moderate robustness; performance degrades in complex or noisy environments.	High robustness; machine learning models can generalize well to new or noisy data with proper training.

Fig 3.2: Performance metrics of existing solutions

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

The objective of this study was to explore and compare various methods for agricultural crop classification using deep learning techniques and other existing solutions. Through the analysis of different approaches, we identified key findings that highlight the strengths, weaknesses, and practical applications of each method.

Key Findings:

[1]. **High Visual Quality and Resolution of Remote Sensing**:
Remote sensing-based solutions (satellite and drone imagery) are highly effective for large-scale crop monitoring, providing detailed and high-resolution images for classifying different crop types. However, the method is costly and dependent on environmental conditions like cloud cover, which can hinder data accuracy.

[2]. **Data Handling and Capacity**:
Deep learning and machine learning methods excel in handling vast amounts of labeled data, which is essential for training robust models. Remote sensing techniques also allow for large-scale data collection, though they require significant computational resources for processing high-resolution imagery.

[3]. **Computational Efficiency**:
Machine learning models, although computationally intensive, provide the best long-term solution for large-scale and complex crop classification tasks, as they can learn patterns from large amounts of labeled data.

In conclusion, **machine learning-based crop classification** methods, particularly those utilizing **deep learning** techniques, provide a highly accurate, robust, and scalable solution for agricultural applications. While remote sensing methods also show great promise for large-scale monitoring, they come with high costs and dependency on environmental factors. Manual and rule-based methods, while useful for specific scenarios, are limited by scalability and computational efficiency.

REFERENCES

REFERENCES

[1]. **Crop classification in high-resolution remote sensing images based on multi-scale feature fusion semantic segmentation model.** https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1196634/full

[2]. **Development of an agricultural crops spectral library and classification of crops.** https://link.springer.com/article/10.1007/s11119-007-9037-x

[3]. https://ieeexplore.ieee.org/abstract/document/6993023/

[4]. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8123d1b6b7f06fef7dac5eca6f313d486620352a

[5]. https://ieeexplore.ieee.org/abstract/document/4812037/

[6]. https://www.tandfonline.com/doi/abs/10.1080/01431160600746456

[7]. https://www.mdpi.com/2072-4292/6/6/5019

[8]. https://www.tandfonline.com/doi/abs/10.1080/01431160310001595046

CONFERENCE/JOURNAL PUBLICATION