DISEASE PREDICTION IN PLANTS AND RECOMMENDATION OF FERTILIZERS USING CNN AND RANDOM FOREST ALGORITHM

A PROJECT REPORT

submitted by

VISHAL D 210701313

YASHVANTH NG 210701319

in partial fulfillment for the award

of the degree of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING

COLLEGE ANNA UNIVERSITY,

CHENNAI MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project report titled **“DISEASE PREDICTION IN PLANTS AND RECOMMENDATION OF FERTILIZERS USING CNN AND RANDOM FOREST ALGORITHM”** is the bonafide work of**“VISHAL D-210701313, YASHVANTH NG -210701319”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr.V.KARTHIK,M.E.,Ph.D.,

PROJECT COORDINATOR

Associate Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai- 602105.

Submitted to Project Viva-Voce Examination held on ________________

Internal Examiner **External Examiner**

ABSTRACT

In our proposed system, a revolutionary strategy for identifying disease caused plants that uses image processing techniques provided by Convolutional neural network as well as fertilizer recommendation for the disease caused plants by using Random forest algorithm is also provided. To maintain plant health, this project utilizes random forest and CNN algorithms. The use of Random Forest algorithms(RFA) with convolutional neural networks (CNNs) in disease prediction and fertilizer recommendation systems will be more efficient in producing efficiency and accuracy. The widespread of disease and poor fertilization techniques damaged agricultural output and food security. CNN algorithms are well known for their image classification and recognition abilities and are especially good at spotting plant illnesses by evaluating leaf images, while Random Forest, an ensemble learning method, is adaptable enough to handle a wide range of agricultural data for fertilizer recommendations. Finally the goal of this approach is to create a disease recognition model with the help of leaf image classification. Based on this the literature and analysis are determined to produce effectiveness, advantages, accuracy, and suitability of both systems for a variety of crops.

ACKNOWLEDGMENT

We thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. MEGANATHAN,B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude and sincere thanks to our beloved Chairperson **Dr.(Mrs).THANGAM MEGANATHAN,Ph.D.,**for her enthusiastic motivation which inspired us a lot in completing this project and we extend our gratitude to our Vice Chairman **Mr.ABHAY SHANKAR MEGANATHAN,B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere and warmful gratitude to our college Principal **Dr. S.N.MURUGESAN,M.E.,Ph.D.**,and **Dr. P. KUMAR,M.E.,Ph.D.**, **Director of Computing and Information Science** and **Head Of the Department of Computer Science and Engineering** and **Project coordinator Dr.V.Karthik,M.E.,Ph.D.**,for his encouragement and guiding us throughout project towards successful completion of project and to our parents, friends, all faculty members,supporting staff for direct and indirect involvement in successful completion of project for their encouragement and support.

VISHAL D

YASHVANTH NG

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

In our proposed system our approach intends to find the disease caused plants by early detection through image processing and also provide suitable fertilizer for that plants . In our proposed solution we use CNN and Random Forest algorithm for the early detection of plants and fertilizer recommendation. CNN are the best algorithm to capture the image and detect the disease in the plant. Random forest algorithm is the best-known algorithm for fertilizer recommendation. The predictions made through this algorithm can produce accuracy. Analyzing agricultural data to produce the best fertilizer recommendations has also shown that Random Forest algorithms work well. The second most prior algorithm is CNN, which is the class of deep learning. It is more accurate in classifying patterns of images to produce the grid-like structure by producing scalability for the project. This helps with the early detection of disease and managing to control plant diseases. This can support farmers in taking precautions to safeguard their crops.

1.1 PROBLEM STATEMENT

Plant diseases can be caused by a variety of factors, and understanding these factors is crucial for effective disease management in agriculture. High levels of moisture and humidity create favorable conditions for the development and spread of fungal and bacterial diseases. Our approach intends To overcome these issues by creating an automated system for plant disease identification. Furthermore, we are utilizing image processing with a Convolution neural network (CNN) for detecting plant disease and then Random Forest Algorithm is used for recommending suitable fertilizer.

1.2 SCOPE OF THE WORK

In the proposed model, The scope of this work involves developing a comprehensive system for plant disease prediction and fertilizer recommendation using a combination of Convolutional Neural Networks (CNN) and Random Forest algorithms. The CNN will be employed to accurately identify and classify plant diseases from leaf images by learning intricate patterns and features. Following disease detection, the Random Forest algorithm will be used to analyze agricultural data to provide precise fertilizer recommendations to the identified plant.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of the proposed system is to develop an integrated system for plant disease prediction and fertilizer recommendation utilizing Convolutional Neural Networks (CNNs) and Random Forest algorithms. This system intends to accurately identify various plant diseases from leaf images and provide tailored fertilizer recommendations based on the diagnosed disease. Our goal of this project is to enhance agricultural productivity, ensure optimal use of resources, and promote sustainable farming practices.

The main objectives of our proposed system are Designing and implementing advanced image processing algorithms that use techniques such as convolutional neural networks (CNNs) to accurately identify and classify disease caused plants based on visual characteristics extracted from images. Collecting and annotating huge plant image datasets to aid model training and validation, resulting in strong performance across a broad botanical species. After diagnosed the disease then we use to integrate with a Random Forest algorithm for recommending appropriate fertilizers based on the identified disease through the class name of the dataset. The ultimate goal is to provide farmers with actionable insights to optimize crop yield and health while minimizing environmental impact.

1.4 RESOURCES

The resources required for the proposed system include high-performance computing infrastructure capable of running Machine learning algorithms efficiently, including GPUs or TPUs for accelerated training and inference. Servers or cloud computing services to host and deploy the image processing and machine learning models. Machine learning frameworks such as TensorFlow, Seaborn, Keras for developing and training convolutional neural networks (CNNs).Image processing libraries like OpenCV2 for preprocessing and feature extraction from plant images. Programming languages such as Python for developing image processing algorithms and machine learning models.

1.5 MOTIVATION

The Disease prediction in plants and Recommendation of fertilizer using CNN and Random forest Algorithm is motivated by a number of essential aspects that overlaps in the plant sector. Accurate and early detection of plant diseases can significantly reduce crop losses, ensuring food security, enhancing agricultural productivity and economic stability for farmers. Meanwhile, Random Forests are robust in handling diverse and complex data for predicting optimal fertilizer requirements based on the identified disease through the class name of the dataset.

CHAPTER 2

2.1 LITERATURE SURVEY

[1] **"The paper titled "Plant Disease Detection Using Image Processing and Machine Learning, 2021" by P. Kulkarni, A. Karwande, T. Kolhe, and S. Kamble** , presents a method for detecting plant diseases through the integration of image processing techniques and machine learning algorithms. This study aims to address the significant challenge of accurately identifying plant diseases in a timely manner, which is crucial for ensuring healthy crop yields and reducing agricultural losses.

[2] **"Deep Learning for Plant Identification and Disease Classification from Leaf Images: Multi-prediction Approaches, 2024" by J. Yao, S.N. Tran, S. Garg, and S. Sawyer** present an advanced deep learning techniques for simultaneously identifying plant species and classifying plant diseases using leaf images. The study aims to enhance the accuracy and efficiency of these critical agricultural tasks through innovative multi-prediction approaches.

[3] **“Hybrid Feature-Based Disease Detection in Plant Leaf Using Convolutional Neural Network, Bayesian Optimized SVM, and Random Forest Classifier, 2022" authors A.K. Singh, S.V.N. Sreenivasu, and U. Mahalaxmi** present an innovative hybrid approach for detecting diseases in plant leaves. This study combines the strengths of Convolutional Neural Networks (CNNs), Bayesian optimized Support Vector Machines (SVMs), and Random Forest classifiers to achieve high accuracy and robustness in disease detection.

[4] **"Tree Species Classification Based on Hybrid Ensembles of a Convolutional Neural Network (CNN) and Random Forest Classifiers," authors U. Knauer, C.S. von Rekowski, M. Stecklina, and T. Krokotsch” 2019**; present a novel approach to classifying tree species by integrating the strengths of Convolutional Neural Networks (CNNs) and Random Forest classifiers. This hybrid ensemble method aims to enhance classification accuracy and robustness by leveraging the complementary capabilities of these two machine learning techniques. The work highlights the advantages of deep learning over traditional image processing methods, particularly in handling the variability in leaf shapes, sizes, and textures.

[5] **"Multi Classification of Tomato Diseases: CNN and Random Forest Hybrid Approach, 2024" authors D. Singh, V. Kumar, and A. Goswami** introduce a hybrid model combining Convolutional Neural Networks (CNNs) and Random Forest classifiers to effectively classify multiple tomato diseases. This study aims to leverage the strengths of both deep learning and ensemble learning techniques to enhance the accuracy and robustness of disease diagnosis in tomato plants.

[6] **"Maize Disease Multi-Classification: Leveraging CNN and Random Forest for Accurate Diagnosis, 2024" by E. Singh, R. Chawla, and R. Kaur** present a hybrid approach that combines Convolutional Neural Networks (CNNs) and Random Forest classifiers to effectively diagnose multiple diseases in maize crops. This study aims to improve the accuracy and reliability of disease detection by utilizing the strengths of both deep learning and ensemble learning methodologies. The authors utilize a comprehensive dataset consisting of images of maize leaves affected by various diseases. The hybrid model is trained and validated on this dataset to ensure its efficacy in diagnosing a range of disease conditions accurately.

[7] **"Identification of Plant-Leaf Diseases Using CNN and Transfer Learning Approach, 2021" by S.M. Hassan, A.K. Maji, M. Jasiński, and Z. Leonowic, "** explore the use of Convolutional Neural Networks (CNNs) combined with transfer learning to accurately identify diseases in plant leaves. This study aims to leverage pre-trained deep learning models to enhance the efficiency and accuracy of disease detection, addressing a critical need in agricultural disease management.

[8] "**Plant Disease and Pest Detection Using Deep Learning-Based Features,2019" M. Türkoğlu and D. Hanbay** present a comprehensive exploration of the application of deep learning techniques for the detection of plant diseases and pests. The literature survey within this paper likely delves into several key areas. Firstly, it may explore existing methodologies and technologies utilized for plant disease and pest detection, ranging from traditional visual inspection methods to more recent computer vision approaches. The review may discuss different approaches to data preprocessing, including image augmentation, normalization. By leveraging these advancements, Sharma and Guleria aim to enhance the diagnostic accuracy and reliability of pneumonia detection systems.

[9] "**Soil Health Analysis and Fertilizer Prediction for Crop Image Identification by Inception-V3 and Random Forest, 2024" by M. Meenakshi and R. Naresh** present a novel approach that combines the Inception-V3 Convolutional Neural Network (CNN) model with Random Forest classifiers to analyze soil health and predict optimal fertilizer usage based on crop images. This study aims to improve agricultural productivity by integrating advanced image analysis and machine learning techniques.

[10] "**Convolutional Recurrent Neural Networks for Hyperspectral Data Classification" published in Remote Sensing,2017" H. Wu and S. Prasad** introduce a novel approach combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to classify hyperspectral data effectively. This study addresses the challenges in processing and analyzing hyperspectral data, which contains rich spectral and spatial information crucial for various remote sensing applications. Effective classification of this data is essential for applications in agriculture, environmental monitoring, and land use mapping.

[11] "**Spectral–Spatial Classification of Hyperspectral Images Using Deep Convolutional Neural Networks, 2015" published in Remote Sensing Letters, authors J. Yue, W. Zhao, S. Mao, and H. Liu** propose an innovative approach that leverages Deep Convolutional Neural Networks (CNNs) for the classification of hyperspectral images. This study focuses on improving the accuracy and efficiency of hyperspectral image classification by utilizing deep learning techniques to simultaneously consider both spectral and spatial information.

[12] "**Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review, 2019" authors A. Signoroni, M. Savardi, A. Baronio, and S. Benini** provide a comprehensive overview of the intersection between deep learning techniques and hyperspectral image analysis. This multidisciplinary review delves into the various applications, methodologies, and challenges associated with employing deep learning in the analysis of hyperspectral images. The authors provide a background on hyperspectral imaging, which captures a wide range of spectral bands across the electromagnetic spectrum. Hyperspectral images contain rich spatial and spectral information, making them valuable for a diverse array of applications in agriculture, environmental monitoring, and remote sensing.

[13] "**Classifying Wheat Hyperspectral Pixels of Healthy Heads and Fusarium Head Blight Disease Using a Deep Neural Network in the Wild Field, 2018" by X. Jin, L. Jie, S. Wang, H. Qi, and S. Li** present an innovative approach for detecting Fusarium head blight (FHB) disease in wheat using hyperspectral imaging and deep neural networks (DNNs). This study addresses the challenge of early disease detection in wheat crops, which is crucial for minimizing yield losses and ensuring food security. The study utilizes hyperspectral imaging to capture detailed spectral information from wheat fields. Hyperspectral data contains hundreds of spectral bands, allowing for the characterization of subtle differences in plant health and disease symptoms.

[14] "**Plant Disease Detection Using Hyperspectral Imaging, 2017" Moghadam, Ward, Goan, Jayawardena, Sikka, and Hernandez"** explore the application of hyperspectral imaging for the detection of plant diseases. Hyperspectral imaging allows for the capture of detailed spectral information across a wide range of wavelengths, enabling the differentiation of healthy and diseased plant tissues based on their spectral signatures. The authors provide an overview of hyperspectral imaging technology, emphasizing its ability to capture fine-grained spectral information from objects or surfaces.

2.2 PROPOSED SYSTEM

DATASET:

The leaf dataset is now hosted in an online repository. The dataset consists of healthy and unhealthy leaf pictures in .jpg format. The images have a resolution of 1600 by 1200 pixels. The dataset consists of 87000 RGB images of healthy and unhealthy plant leaves having 38 classes out of which We have selected only 6 classes for experimentation of our algorithm: Potato(Healthy), Potato(Early Blight), Potato(Late_blight), Rice(Healthy), Rice(Rust), Rice(Powdery). In regard to the studies , the dataset's makeup is divided by the proportion. Plant Village, curated by Sharada P. Mohanty and colleagues, is a notable platform dedicated to plant health. This dataset would enable the development of accurate and robust predictive models capable of diagnosing plant diseases and recommends fertilizer interventions to optimize crop health and productivity. It serves as a comprehensive resource for farmers, researchers, and enthusiasts, offering valuable information and tools for the diagnosis and management of plant diseases. (accessed at 14:27 on September 21,2022, from https://paperswithcode.com/dataset/plantvillage)80% of the data were used for training, 10% for validation, and 10% for testing.

MODEL ARCHITECTURE:

The architecture model and the layers that make up the architecture, which are Conv2D, MaxPooling2D, Flattern, and Dense Layer A parameter value and an output shape are generated by each layer. The proposed model architecture is more sophisticated than the previous and uses a pretrained model VGG16 architecture, with higher parameter values and an output shape. Our model first performs the convolution process and filters the input image by resizing its dimensions to 128 x 128. Following convolution, the Pooling Layer is applied, and if it is complete, the subsequent convolution is applied, changing the dimensions and allowing the Pooling Layer to continue.

TRAINING AND TESTING:

Our Proposed model learns to recognize various plant species to ensure the accuracy and reliability of the predictive model. Initially, a comprehensive dataset comprising images of plants affected by various diseases and fertilizer records, is collected and preprocessed. This dataset is then divided into training and testing sets, with the training set used to train the CNN and Random Forest models. During training, the CNN learns to extract spatial features from the plant images, while the Random Forest model learns to classify diseases and recommend appropriate fertilizers based on the Identified disease through the class name of the dataset. Through iterative refinement and validation, the disease prediction and fertilizer recommendation model achieves high accuracy and reliability.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.2 SYSTEM ARCHITECTURE DIAGRAM

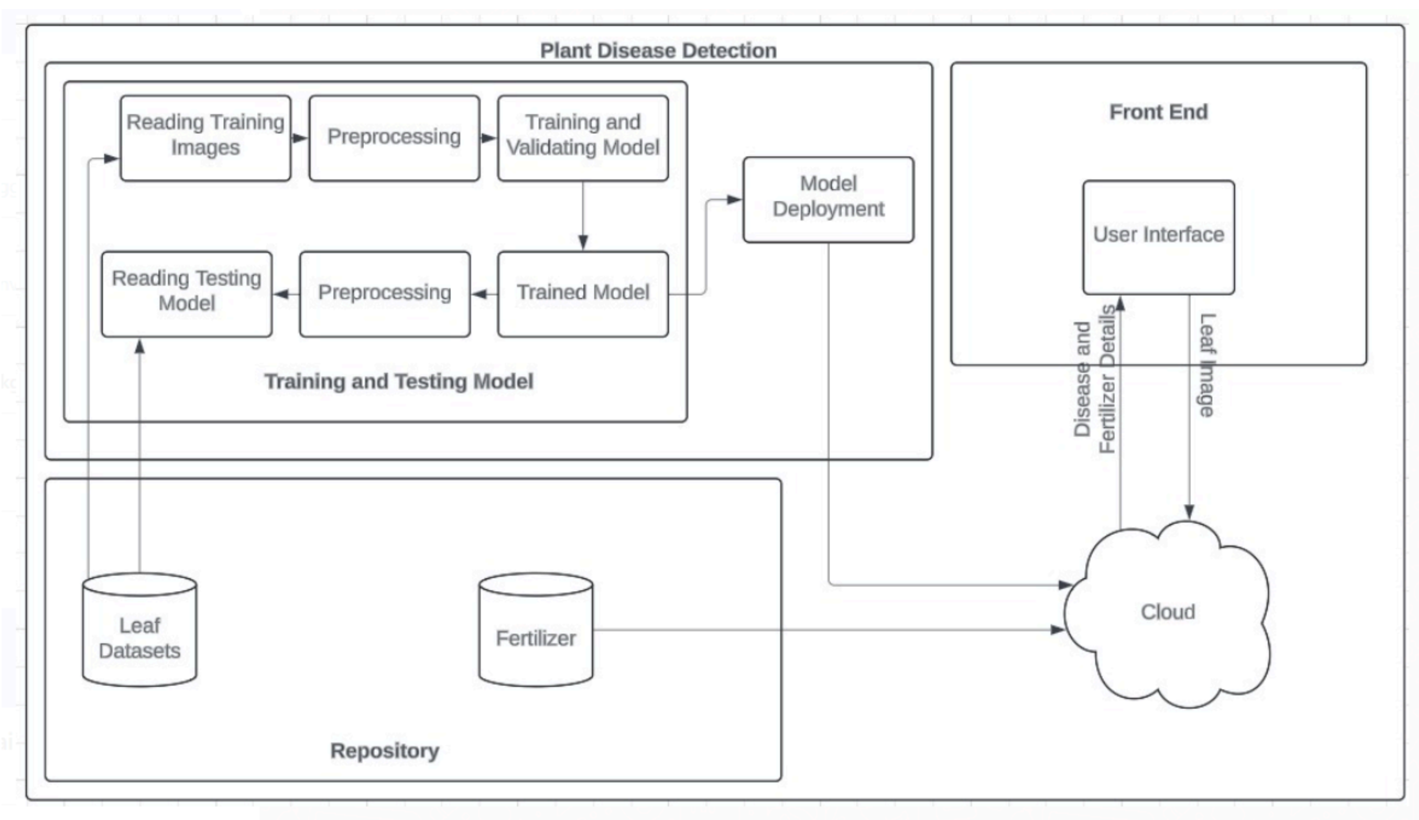

Fig 3.2.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

Table 3.3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification.The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product.

Table 3.3.2 Software Requirements

S.NO	REQUIREMENT
1	Jupyter Notebook
2	StreamLit API
3	TensorFlow

3.4 DESIGN OF THE ENTIRE SYSTEM:

3.4.1 SEQUENCE DIAGRAM:

A sequence diagram simply depicts the interaction between the objects in a sequential order. An sequence diagram is used to show the interactive behavior of a system. The sequence diagram for Disease prediction in plants and recommendation of fertilizer is attached in the below figure 3.4.1.

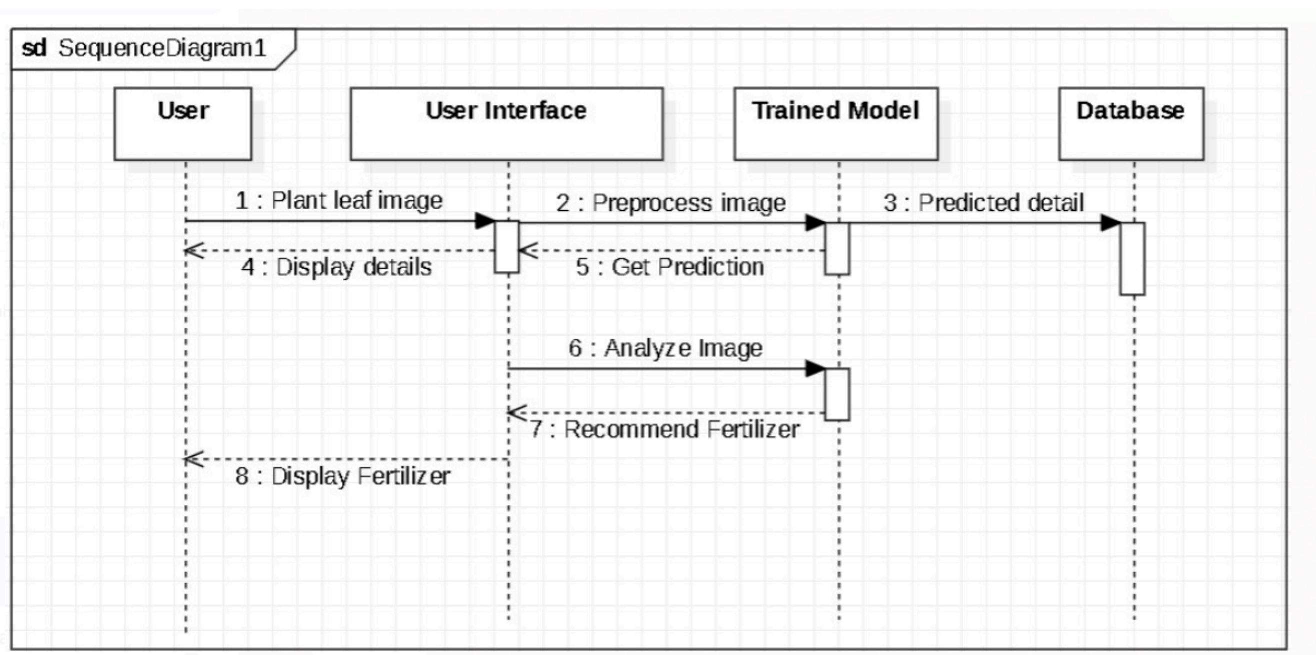

Fig 3.4.1: Sequence Diagram

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGY

There are numerous critical elements in the methodology for identifying disease plants through image processing employing deep learning techniques with datasets collected from various sources. These images are labeled with the respective plant types. The image dataset will undergo preprocessing, which also includes data augmentation techniques like rotation, scaling, and color adjustments to enhance the diversity of the training data and remove any irrelevant or low-quality photos from the dataset to improve quality. For feature extraction and classification, use appropriate deep learning models such as convolutional neural networks (CNNs). Use transfer learning with pre-trained models to take advantage of their learned representations and improve performance. Develop a user-friendly application that enables users to upload images to receive real-time diagnosis and recommendation of fertilizers for the respective by integrating it into the trained model. The model will be continuously updated with data to enhance its accuracy.

4.2 MODULE DESCRIPTION

The Disease prediction in plant and Recommendation of fertilizers using CNN and Random forest algorithm is divided into three major sections. First, the Data Collection and Preprocessing Module collects a broad dataset of healthy and infected plant photos and improves their quality using preprocessing techniques such as normalization and augmentation. The Training model such as CNN is used to train the dataset to correctly identify the infected plant species. Random Forest algorithm is used for recommending appropriate fertilizers based on the identified disease through the class name of the dataset.

Furthermore, the disease prediction with fertilizer recommendation module offers user-friendly interfaces and seamless integration with agricultural management systems, allowing farmers to access the platform conveniently from their desktop or mobile devices. Through intuitive dashboards and interactive features, users can upload images of diseased plants, receive instant disease predictions, and obtain personalized fertilizer recommendations based on real-time data analysis. By harnessing the power of advanced technologies and user-centric design principles, this module empowers farmers with actionable insights and decision support tools to effectively manage plant diseases and optimize fertilizer usage, ultimately contributing to sustainable agriculture and food security.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working application.

Plant disease prediction Webpage

Fig 5.1.1: Plant Disease prediction website

Output from predicting the image

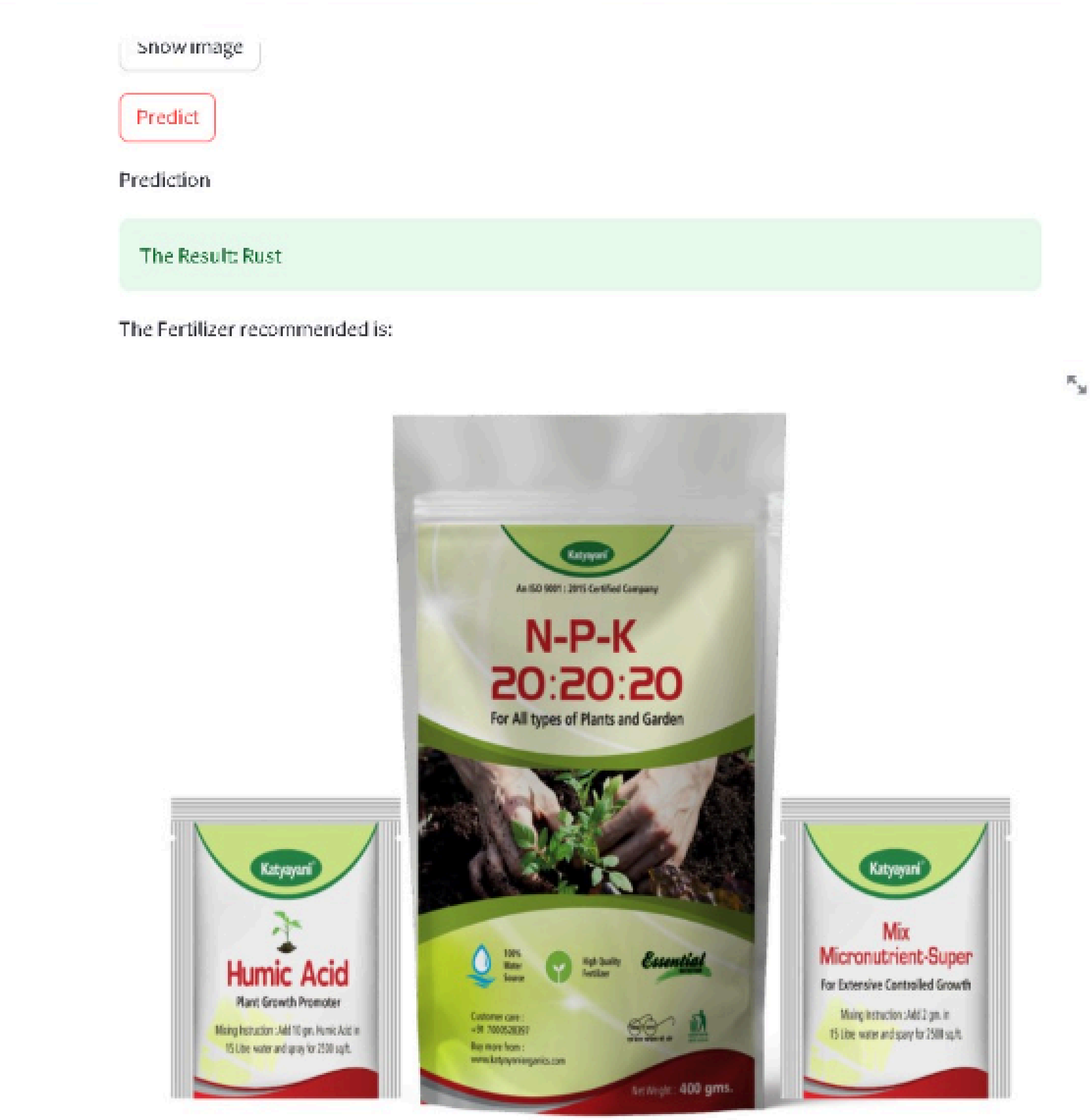

Fig 5.1.2: Output of predicted plant

Confusion Matrix :

The plant disease prediction trained model is evaluated and the confusion matrix for the trained model is attached in below Figure 5.1.3

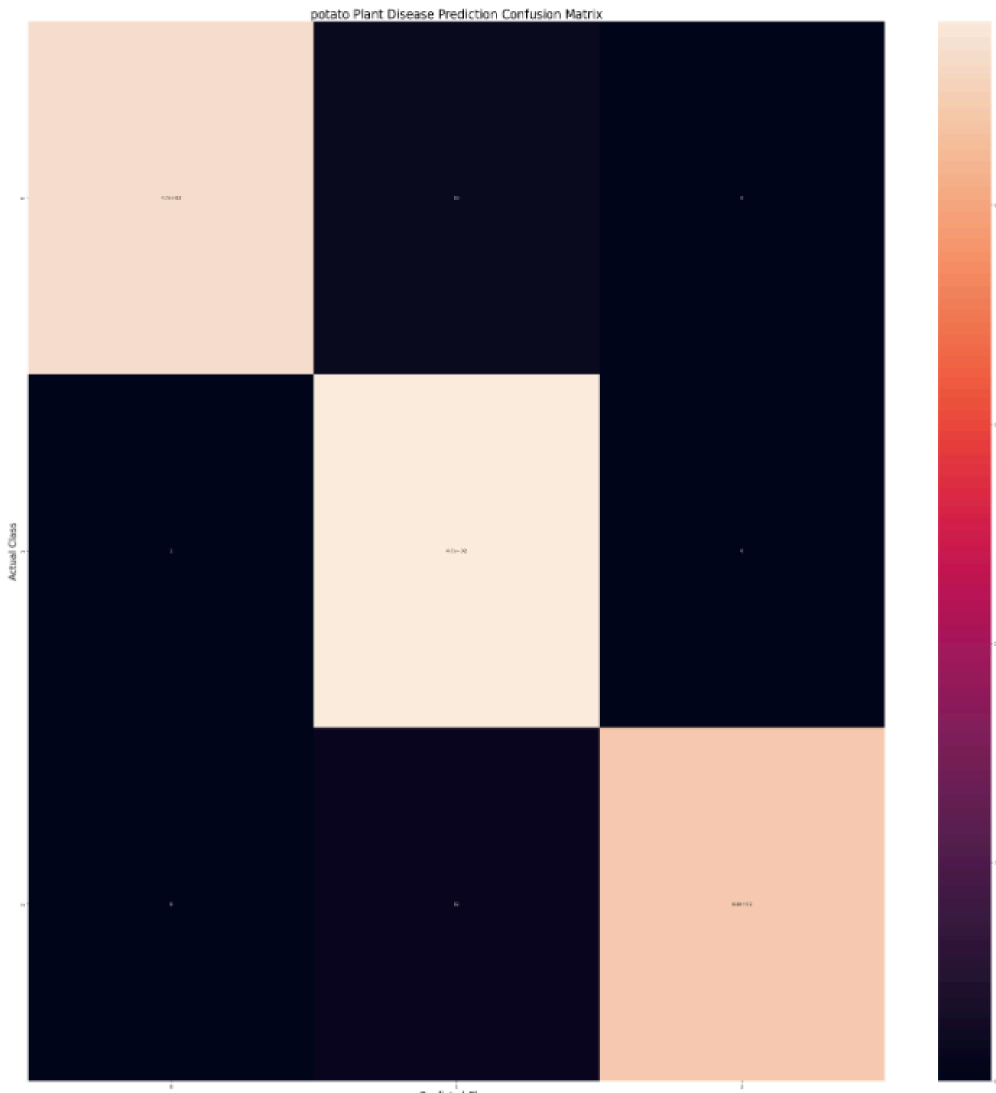

Fig 5.1.3: Confusion matrix

Training and Testing Accuracy Graph:

The proposed model is evaluated and the testing and training accuracy graph is obtained. Spliting the dataset into training and validation sets (e.g., 80-20 split). Training the model using the training set, adjusting hyperparameters to optimize performance. Employ techniques such as dropout and batch normalization to prevent overfitting. The training and testing accuracy rate of the model is attached in the below figure 5.1.4

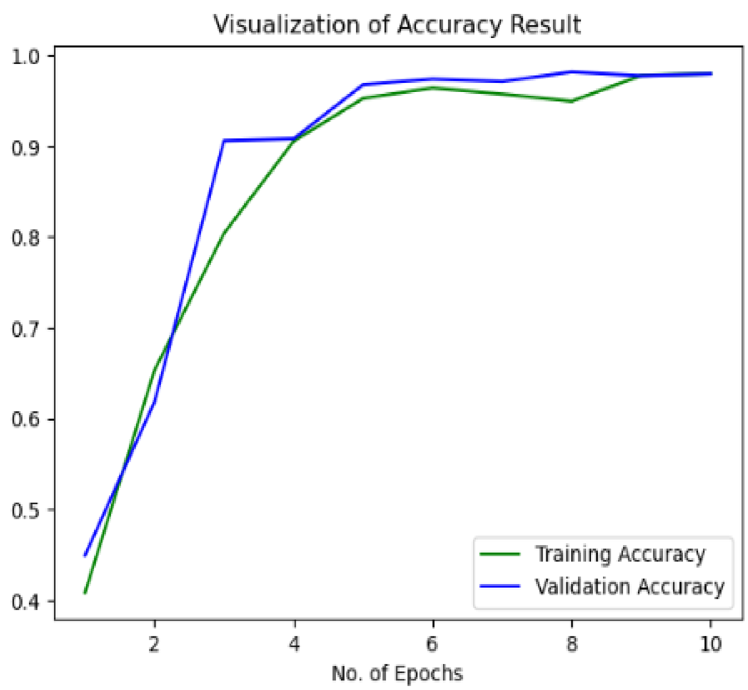

Fig 5.1.4: Training and Testing Accuracy Graph

5.2 RESULT

The implementation of a Disease prediction in plants and recommendation of fertilizers using CNN and Random forest algorithm represents a remarkable journey by achieving 98% accuracy in identifying disease caused plants. Also the proposed system is computationally efficient because of the use of statistical image processing and machine learning model. The created system, which employs advanced image processing techniques and Machine learning algorithms.

The Proposed model enables farmers to make data-driven decisions that optimize yield potential while minimizing the risk of disease outbreaks and nutrient deficiencies. Moreover, the automated nature of the system streamlines the decision-making process, allowing for timely interventions and ultimately contributing to enhanced agricultural productivity and sustainability. As technology continues to advance, the integration of CNNs and Random Forests holds immense potential for revolutionizing crop management practices and empowering farmers with the tools needed to navigate the complexities of modern agriculture.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The proposed model is an important step forward in the fields of plant disease prediction and its management. Finally the goal of this approach is reached by creating a disease recognition model with the help of leaf image classification. Based on this the literature and analysis are determined to produce effectiveness, advantages, accuracy, and suitability of both systems for a variety of crops. The created approach has exhibited extraordinary accuracy, reaching a 98% identification rate for infected plants using visual characteristics retrieved from photos. Through accurate disease prediction, farmers can proactively mitigate the impact of diseases on crop yields, while targeted fertilizer recommendations ensure optimal nutrient management, leading to improved productivity and sustainability in agriculture.

6.2 FUTURE ENHANCEMENT

Real Time Monitoring: Connect the identification system to Internet of Things (IoT) devices, such as smart cameras or sensors, deployed throughout the supply chain.

Authenticity and accesibility: Create user-friendly mobile apps or web platforms that allow customers to check the authenticity of Healthy plant items by scanning QR codes or submitting photographs.

APPENDIX

main_app.py:

```
# Library imports

import numpy as np

import streamlit as st

import cv2

from keras.models import load_model

import tensorflow as tf



# Loading the Model

model = load_model('medicinal_plants_cnn.h5')



# Name of Classes
CLASS_NAMES = ('Corn_(maize)___Common_rust_',
'Corn_(maize)___Northern_Leaf_Blight', 'Corn_(maize)___healthy',
              '  'Potato___Early_blight', 'Potato___Late_blight', 'Potato___healthy'
')



# Setting Title of App

st.title("Plant Disease

Detection")

st.markdown("Upload an image of the plant leaf")



# Uploading the dog image
```

```
plant_image = st.file_uploader("Choose an image...", type = "jpg")
submit = st.button('PREDICT PLANT')


# On predict button click
if submit:
    if plant_image is not None:
        # Convert the file to an opencv image.
        file_bytes = np.asarray (bytearray(plant_image.read()), dtype = np.uint8)
        opencv_image = cv2.imdecode(file_bytes, 1)


        # Displaying the image
        st.image(opencv_image, channels="BGR")
        st.write(opencv_image.shape)
```

Resizing the image

```
opencv_image = cv2.resize(opencv_image, (150, 150))

# Convert image to 4 Dimension
opencv_image.shape = (1, 150, 150, 3)

#Make Prediction
Y_pred = model.predict(opencv_image)
result = CLASS_NAMES[np.argmax(Y_pred)]
st.title(str("The predicted plant is " + result))
```

Infected_plants.py:

```
import pandas as
pd import numpy
as np import os
import matplotlib as plt
from sklearn.datasets import load_files

#The path of our data on drive
data_dir = r'C:\Users\KB VIMAL\Desktop\infected plants’
```

```
#Loading our Data

data = load_files(data_dir)


folders=os.listdir(r" ClC:\Users\KB VIMAL\Desktop\infected

plants\early_blight\dataset1")

print(folders)

X = np.array(data['filenames'])

y = np.array(data['target'])

labels = np.array(data['target_names'])

x_tr=97

x_ts=1

print('Data files - ',X)

print('Target labels - ',y)

print('Number of training files : ', X.shape[0])

print('Number of training targets : ', y.shape[0])


from keras.preprocessing.image import img_to_array, load_img

def

    convert_img_to_arr(file_path_list):

    arr = []

    #size=64,64
```

```
    img_width, img_height = 150,150
    for file_path in file_path_list:
        img = load_img(file_path, target_size = (img_width, img_height))
        img = img_to_array(img)
        arr.append(img)
        #arr.append(cv2.resize(img,size))
    return arr

X = np.array(convert_img_to_arr(X))
print(X.shape)
print('First training item : ',X[0])

import matplotlib.pyplot as plt

fig = plt.figure(figsize = (16,9))
for i in range(5):
    ax = fig.add_subplot(1,5,i+1,xticks=[],yticks=[])
    ax.imshow((X[i].astype(np.uint8)))
    plt.title(folders[y[i]])

# Let's resize or rescale training data
```

```
X = X.astype('float32')/255

# Let's confirm the number of classes :)
no_of_classes = len(np.unique(y))
no_of_classes

from tensorflow.python.keras.utils import np_utils

# let's converts a class vector (integers) to binary class matrix.
y = np_utils.to_categorical(y-1,no_of_classes)
y[0]

from sklearn.model_selection import train_test_split

# let's splite the data into subsets and explore their shapes !

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
print('The test Data Shape ', X_test.shape[0])
X_test, X_valid, y_test, y_valid = train_test_split(X_test,y_test, test_size = 0.5)
print('The training Data Shape ', X_valid.shape[0])
```

```
import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPool2D, Dense, Flatten,

Dropout from tensorflow.keras.preprocessing.image import

ImageDataGenerator from keras.callbacks import ReduceLROnPlateau


model = Sequential()


model.add(Conv2D(filters=64, kernel_size=(3,3), padding='same',

input_shape=X_train.shape[1:], activation='relu', name='Conv2D_1'))

model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu',

name='Conv2D_2')) model.add(MaxPool2D(pool_size=(2,2), name='Maxpool_1'))

model.add(Dropout(0.5))


model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same', activation='relu',

name='Conv2D_3'))

model.add(Conv2D(filters=128, kernel_size=(3,3), activation='relu', name='Conv2D_4'))

model.add(MaxPool2D(pool_size=(2,2), name='Maxpool_2'))

model.add(Dropout(0.5))
```

```
model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same', activation='relu',
name='Conv2D_5'))
model.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu', name='Conv2D_6'))
model.add(MaxPool2D(pool_size=(2,2), name='Maxpool_3'))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(units=512,                    activation='relu',
name='Dense_1')) model.add(Dropout(0.5))
model.add(Dense(units=128,                    activation='relu',
name='Dense_2'))                  model.add(Dense(units=128,
activation='relu',                            name='Dense_3'))
model.add(Dense(units=128,                    activation='relu',
name='Dense_4'))

model.add(Dense(units=no_of_classes, activation='softmax', name='Output'))

from keras.optimizers import RMSprop

optimizer = RMSprop(learning_rate=0.001, rho=0.9, epsilon=1e-08, decay=0.0)

model.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

```
import time
from keras.callbacks import ModelCheckpoint, EarlyStopping


# Time to train our model !
epochs = 100
batch_size=128


train_datagen = ImageDataGenerator(
        rotation_range=10,
        zoom_range = 0.1,
        width_shift_range=0.1,
        height_shift_range=0.1,
        horizontal_flip=True)


test_datagen = ImageDataGenerator()


train_generator = train_datagen.flow(
    X_train,y_train,
    batch_size=batch_size)


validation_generator = test_datagen.flow(
```

```
    X_valid,y_valid,
    batch_size=batch_size)

checkpointer = ModelCheckpoint(filepath = r"C:\Users\vinot\My ML Projects\Medicinal
Plants Classifcation\mediplant.keras", save_best_only = True, verbose = 1)
learning_rate_reduction=ReduceLROnPlateau(monitor='val_accuracy', patience = 3, verbose
= 1, factor = 0.5, minlr = 0.00001)



start = time.time()

# let's get started !

history=model.fit(train_generator,
                  epochs=epochs,
                  validation_data = validation_generator,
                  verbose=1,
                  steps_per_epoch=len(X_train) // batch_size,
                  #validation_steps=len(X_valid) //batch_size,
                  # callbacks=[checkpointer, learning_rate_reduction]
          )
```

```
end = time.time()

duration = end - start
(eval_loss, eval_accuracy) = model.evaluate(
    X_test, y_test, batch_size=16, verbose=2)
print ('\n This Model took %0.2f seconds (%0.1f minutes) to train for %d epochs'%(duration,
duration/60, epochs) )

print("Accuracy: {:.2f}%".format(eval_accuracy * 100))
print("Loss: {}".format(eval_loss))

print('The train Data Shape ', X_test.shape[1:])
x = np.asarray(X_valid)
images = np.vstack([x])
classes = model.predict(images)
print('Predicted class is :')
print((classes))
# Finding max value from predition list and comaparing original value vs predicted
from sklearn.metrics import confusion_matrix
print("Originally : ", folders[np.argmax(y_valid[11])])
```

```
print("Predicted : ", folders[np.argmax(classes[11])])

print(max(y_pred[11]))


# from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

import seaborn as sns

# Predict the values from the validation dataset

Y_pred = model.predict(X_train)

# Convert predictions classes to one hot vectors

Y_pred_classes = np.argmax(Y_pred,axis = 1)

# Convert validation observations to one hot vectors

Y_true = np.argmax(y_train,axis = 1)

# compute the confusion matrix

confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)

# plot the confusion matrix

f,ax = plt.subplots(figsize=(15,15))

sns.heatmap(confusion_mtx, annot=True, linewidths=0.01,cmap="Greens",linecolor="gray",

fmt= '.1f',ax=ax)

plt.xlabel("Predicted

Label") plt.ylabel("True

Label") plt.title("Confusion
```

```
plt.show()


import matplotlib.pyplot as plt

acc_train =

history.history['accuracy']

acc_val = history.history['val_accuracy']

epochs = range(1,11)

plt.plot(epochs,acc_train,'g',label='Training Accuracy')

plt.plot(epochs,acc_val,'b',label='Testing Accuracy') # validation

accuravy plt.title('Training and Testing Accuracy')# validation accuravy

plt.xlabel('Epochs')

plt.ylabel('Accuracy'

) plt.legend()

plt.show()


import matplotlib.pyplot as plt

loss_train = history.history['loss']

loss_val = history.history['val_loss']
```

```
model.add(Flatten())

model.add(Dense(units=512,                        activation='relu',

name='Dense_1')) model.add(Dropout(0.5))

model.add(Dense(units=128,                        activation='relu',

name='Dense_2'))                    model.add(Dense(units=128,

activation='relu',                              name='Dense_3'))

model.add(Dense(units=128,                        activation='relu',

name='Dense_4'))



plt.plot(epochs,loss_train,'g',label='Training loss')

plt.plot(epochs,loss_val,'b',label='Testing loss')# validation loss

plt.title('Training and Testing loss')# validation loss

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()


print('The train Data Shape ', X_test.shape[1:])

x = np.asarray(X_valid)
```

```
images = np.vstack([x])

classes = model.predict(images)
```

```
print('Predicted class is :')

print((classes))

# Finding max value from predition list and comaparing original value vs predicted

from sklearn.metrics import confusion_matrix

print("Originally : ", folders[np.argmax(y_valid[11])])

print("Predicted : ", folders[np.argmax(classes[11])])

print(max(y_pred[11]))



model.save('plant_disease_predicted_model_cnn1.h5')
```

REFERENCES

[1] “**The paper titled "Plant Disease Detection Using Image Processing and Machine Learning” by** P. Kulkarni,and S. Kamble(published in **2021)**

[2] "**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**" by Aurélien Géron (Published in **2019**)

[3] "**Python Machine Learning**" by Ivan Vasilev and Daniel Slater in **2017.**

[4] **"Deep Learning with Python**" by Francois Chollet (Published in **2017**).

[5] "**Practical Python and OpenCV**" by Adrian Rosebrock (Published in **2019**)

[6] “**Machine Learning Yearning**" by Andrew Ng (Published in **2018**)

[7] "**Python Deep Learning Projects**" by Matthew Lamons, Rahul Kumar, and Abhishek Nagaraja (Published in **2019**)

[8] “**ImageNet classification with deep convolutional neural networks." In Advances in neural information processing systems** " by Krizhevsky, A., Sutskever, I., & Hinton, G. E. (Published in **2012**)

[9] " **Classification and regression by randomForest "** by Liaw, A., & Wiener, M (Published in **2002**)

[10] " **Random forests, Machine learning,** " by Breiman, L (Published in **2001**)