Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING
1156CS701- MAJOR PROJECT(INHOUSE)
WINTER SEMESTER 2023-2024
REVIEW - II

"Plant Disease Detection and Prevention using CNN"

SUPERVISED BY

Mr. Ignatious K Pious, ME.,
ASSISTANT PROFESSOR

PRESENTED BY

1. GUTHA SAINATH REDDY (17707) (20UECS0368)
2. AMUDALA BHANU TEJA (18300) (20UECS0046)
3. AMUDALA POOJITHA (18301) (20UECS0047)

AGENDA

- ABSTRACT
- OBJECTIVE
- INTRODUCTION
- LITERATURE REVIEW (SOFT COPY OF PAPERS TO BE LINKED AS HYPERLINK)
- DESIGN AND METHODOLOGIES
- IMPLEMENTATION
- TESTING
- INPUT AND OUTPUT
- INCLUDE DEMO VIDEO-1 (Till REVEW-1)
- INCLUDE DEMO VIDEO-2(Complete Implementation of Project)
- CONCLUSION
- WEB REFERENCES LINK (TILL REVIEW DATE ALL LINKS TO BE INCLUDED DAY WISE)
- PLAGIARISM REPORT OF PPT
- REFERENCES

ABSTRACT

Plant diseases pose a significant threat to global food security, often leading to catastrophic losses in agricultural productivity. These diseases not only diminish the quality and quantity of agricultural products but can also result in complete crop failure in severe cases. Among the plethora of plant diseases, Leaf Spot stands out as a particularly troublesome ailment, affecting various crops at different growth stages and causing substantial yield losses. Efficient detection and prevention methods are crucial for mitigating the impact of plant diseases. In recent years, deep learning, especially Convolutional Neural Networks (CNNs), has emerged as a promising tool for disease detection due to its remarkable performance in image analysis tasks. In this study, we focused on leveraging the capabilities of deep learning, particularly the EfficientNetV2 model, for the detection of Leaf Spot in plants. The methodology employed in our study involved training the EfficientNetV2 model to recognize multiscale features associated with Leaf Spot disease. By utilizing CNNs, we were able to effectively remove unwanted background noise from input images, enhancing the accuracy of disease detection. A comprehensive set of experiments was conducted to evaluate the performance of our approach and compare it with existing models such as EfficientNet and traditional CNNs. The experimental results revealed that our proposed approach achieved an impressive detection accuracy of 93.26%. This high level of accuracy is a testament to the effectiveness of deep learning techniques in plant disease detection. Such precise identification of Leaf Spot can enable farmers to take timely and targeted measures to mitigate the spread of the disease, thereby minimizing yield losses and ensuring food security. The implications of our findings extend beyond academic research, offering practical solutions for enhancing agricultural productivity and food safety. By integrating deep learning-based disease detection systems into agricultural practices, farmers can make informed decisions regarding pest management, resource allocation, and crop protection strategies. Furthermore, the scalability and adaptability of our approach make it suitable for deployment in diverse agricultural settings worldwide. Looking ahead, there are several avenues for future research and improvement. Fine-tuning the model architecture and optimizing hyperparameters could potentially further enhance detection accuracy. Additionally, exploring real-time monitoring systems and integrating remote sensing technologies could facilitate early disease detection and intervention, thereby offering a proactive approach to plant disease management. In conclusion, our study underscores the significance of deep learning in revolutionizing plant disease detection and prevention efforts. By harnessing the power of EfficientNetV2 and CNNs, we have demonstrated a robust and efficient approach for combating Leaf Spot and safeguarding global food production.

OBJECTIVES

Aim of the Project:

Modern image processing and deep learning-based techniques are widely used for the detection of plant leaf disease. Many diagnostic methods use a Convolutional Neural Network (CNN) and a pre-trained model to detect and classify healthy and unhealthy plants, used segmentation to remove background data and applied a trained neural network for classification.

Scope of the Project:

- Zhang et al. proposed a method to diagnose cucumber plant diseases by separating images with diseased patches by combining K-means, exploring the condition and color of infected leaf lesions, and separating unhealthy leaf images using scant resentment.
- Proposed method improved Ocular-head deep neural networks for classification by reading the feature maps highlighting the essential regions that also weaken the meaningless connected layers. Various artificial intelligence and image-based plant disease detection approaches have been proposed.

INTRODUCTION

- Plant diseases refer to various abnormal conditions, disorders, or pathogens that adversely affect the health and vitality of plants. These diseases can manifest in numerous ways, including physical damage, poor growth, reduced yield, and even plant death.
- They pose significant challenges to agricultural, horticultural, and forestry practices worldwide. Pathogens are microorganisms such as bacteria, fungi, viruses, and nematodes that invade plant tissues, disrupting normal cellular functions.
- They may spread through soil, water, air, or by vectors like insects. Abiotic factors like extreme temperatures, drought, excessive moisture, or pollution can weaken plants, making them more susceptible to diseases. Some plants are genetically predisposed to specific diseases.
- Breeding programs aim to develop disease-resistant varieties. Poor cultural practices, such as overcrowding, improper watering, or inadequate nutrition, can stress plants and create conditions favorable for diseases.
- Preventing plant diseases is essential for maintaining healthy crops and ecosystems. Avoid planting the same crop in the same location year after year to break the life cycle of pathogens

INTRODUCTION

- Choose plant varieties bred for resistance to prevalent diseases in your region. Keep planting areas clean by removing diseased plant material, including fallen leaves and fruits.
- Avoid overwatering, which can create conditions conducive to fungal diseases, and water plants at the base to minimize leaf wetness.
- Plant With adequate spacing to promote airflow and reduce humidity around plants. Maintain healthy soil through organic matter addition and proper pH adjustments.
- Preventing and managing plant diseases requires vigilance and a combination of strategies tailored to specific plant types, diseases, and environmental conditions.
- A proactive approach is essential to protect plant health and ensure robust agricultural and horticultural practices.

LITERATURE REVIEW

PAPER1:
TITLE: Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern
AUTHOR: KHALID M. HOSNY, WALAA M. EL-HADY, FARID M. SAMY2 , ELENI VROCHIDOU, GEORGEA.PAPAKOSTAS
YEAR: 2023
LINK: **https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10153610**

ABSTRACT:

Plant diseases are one of the primary causes of decreased agricultural production quality and quantity. With ongoing changes in plant structure and cultivation techniques, new diseases are constantly arising on plant leaves. Thus, accurate classification and detection of plant leaf diseases in their early stages will limit the spread of the infection and support the healthy development of plant production. This work proposes a novel lightweight deep convolutional neural network (CNN) model for obtaining high-level hidden feature representations. The deep features are then fused with traditional handcrafted local binary pattern (LBP) features to capture local texture information in plant leaf images. The proposed model is trained and tested on three publicly available datasets (Apple Leaf, Tomato Leaf, and Grape Leaf). On the three datasets, the proposed approach achieves 99%, 96.6%, and 98.5% validation accuracies, respectively, and 98.8%, 96.5%, and 98.3% test accuracies, respectively. The results of the experiments show that the proposed approach can provide a better control solution for plant diseases.

LITERATURE REVIEW

PAPER2:

TITLE: Machine Learning and Deep Learning for Plant Disease Classification and Detection

AUTHOR: VASILEIOS BALAFAS, EMMANOUIL KARANTOUMANIS, MALAMATI LOUTA, NIKOLAOS PLOSKAS

YEAR: 2023

LINK : https://users.uowm.gr/louta/JOURNALS/J32.pdf

ABSTRACT:

Precision agriculture is a rapidly developing field aimed at addressing current concerns about agricultural sustainability. Machine learning is the cutting edge technology underpinning precision agriculture, enabling the development of advanced disease detection and classification methods. This paper presents a review of the application of machine learning and deep learning techniques in precision agriculture, specifically for detecting and classifying plant diseases. We propose a novel classification scheme that categorizes all relevant works in the associated classes. We separate the studies into two main categories depending on the methodology that they use (i.e., classification or object detection). In addition, we present the available datasets for plant disease detection and classification. Finally, we perform an extensive computational study on five state-of-the-art object detection algorithms on Plant Doc dataset to detect diseases present on the leaves.

LITERATURE REVIEW

PAPER3:

TITLE: Handling Severity Levels of Multiple Co-Occurring Cotton Plant Diseases Using Improved YOLOX Model

AUTHOR: SEROSH KARIM NOON, MUHAMMAD AMJAD, MUHAMMAD ALI QURESHI, ABDUL MANNAN

YEAR: 2022

LINK : **https://www.researchgate.net/figure/Details-of-Cotton-disease-severity-dataset-CoSev-before-augmentation_tbl1_366623463**

ABSTRACT:

Automatic detection of plant diseases has emerged as a challenging field in the last decade. Computer vision-based advancements have helped in the timely and accurate identification of diseases, making possible an appropriate treatment and hence ensuring an increased yield. Diseases attack in different formations on a plant; the most severe being multiple diseases appearing on a single leaf. Moreover, as various diseases progress, they generate similar-looking symptoms making the task of identification further difficult. This work addresses these two problems with the help of an improved YOLOX model. We propose a modified Spatial Pyramid Pooling (SPP) layer to effectively extract relevant features at various scales from the training data.

LITERATURE REVIEW

PAPER4:
TITLE: A Systematic Literature Review on Plant Disease Detection: Motivations, Classification Techniques, Datasets, Challenges, and Future Trends
AUTHOR: WASSWA SHAFIK, , ALI TUFAIL, ABDALLAH NAMOUN ,LIYANAGE CHANDRATILAK DE SILVA, ROSYZIE ANNA AWG HAJI MOHD APONG
YEAR: 2023
LINK : https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10147225
ABSTRACT:

Plant pests and diseases are a significant threat to almost all major types of plants and global food security. Traditional inspection across different plant fields is time-consuming and impractical for a wider plantation size, thus reducing crop production. Therefore, many smart agricultural practices are deployed to control plant diseases and pests. Most of these approaches, for example, use vision-based artificial intelligence (AI), machine learning (ML), or deep learning (DL) methods and models to provide disease detection solutions. However, existing open issues must be considered and addressed before AI methods can be used. In this study, we conduct a systematic literature review (SLR) and present a detailed survey of the studies employing data collection techniques and publicly available datasets. To begin the review, 1349 papers were chosen from five major academic databases, namely Springer, IEEE Xplore, Scopus, Google Scholar, and ACM library.

LITERATURE REVIEW

PAPER5:
TITLE: Cardamom Plant Disease Detection Approach Using EfficientNetV2
AUTHOR: SUNIL C. K., JAIDHAR C. D., NAGAMMA PATIL
YEAR: 2022
LINK : https://ieeexplore.ieee.org/document/9663367?denied=
ABSTRACT:

Cardamom is a queen of spices. It is indigenously grown in the evergreen forests of Karnataka, Kerala, Tamil Nadu, and the northeastern states of India. India is the third largest producer of cardamom. Plant diseases cause a catastrophic influence on food production safety; they reduce the eminence and quantum of agricultural products. Plant diseases may cause significantly high loss or no harvest in dreadful cases. Various diseases and pests affect the growth of cardamom plants at different stages and crop yields. This study concentrated on two diseases of cardamom plants, Colletotrichum Blight and Phyllosticta Leaf Spot of cardamom and three diseases of grape, Black Rot, ESCA, and Isariopsis Leaf Spot. Various methods have been proposed for plant disease detection, and deep learning has become the preferred method because of its spectacular accomplishment. In this study, U2 -Net was used to remove the unwanted background of an input image by selecting multiscale features. This work proposes a cardamom plant disease detection approach using the EfficientNetV2 model. A comprehensive set of experiments was carried out to ascertain the performance of the proposed approach and compare it with other models such as EfficientNet and Convolutional Neural Network (CNN). The experimental results showed that the proposed approach achieved a detection accuracy of 98.26%.

Sl. no	AUTHOR	YEAR	TITLE	THEME
1.	KHALID M. HOSNY, WALAA M. EL-HADY, FARID M. SAMY2 , ELENI VROCHIDOU, GEORGE A. PAPAKOSTAS	2023	Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern	This work proposes a novel lightweight deep convolutional neural network (CNN) model for obtaining high-level hidden feature representations.
2.	VASILEIOS BALAFAS, EMMANOUIL KARANTOUMANIS, MALAMATI LOUTA, NIKOLAOS PLOSKAS	2023	Machine Learning and Deep Learning for Plant Disease Classification and Detection	This paper presents a review of the application of machine learning and deep learning techniques in precision agriculture, specifically for detecting and classifying plant diseases.
3.	SEROSH KARIM NOON, MUHAMMAD AMJAD, MUHAMMAD ALI QURESHI, ABDUL MANNAN	2022	Handling Severity Levels of Multiple Co-Occurring Cotton Plant Diseases Using Improved YOLOX Model	We propose a modified Spatial Pyramid Pooling (SPP) layer to effectively extract relevant features at various scales from the training data.
4.	WASSWA SHAFIK, , ALI TUFAIL, ABDALLAH NAMOUN ,LIYANAGE CHANDRATILAK DE SILVA, ROSYZIE ANNA AWG HAJI MOHD APONG	2023	A Systematic Literature Review on Plant Disease Detection: Motivations, Classification Techniques, Datasets, Challenges, and Future Trends	we conduct a systematic literature review (SLR) and present a detailed survey of the studies employing data collection techniques and publicly available datasets
5.	SUNIL C. K., JAIDHAR C. D., NAGAMMA PATIL	2022	Cardamom Plant Disease Detection Approach Using EfficientNetV2	This work proposes a cardamom plant disease detection approach using the EfficientNetV2 model.

DESIGN AND METHODOLOGIES

LIST OF MODULES

- Dataset Collection
- Data Preprocessing
- Feature Extraction
- Model Selection and Metrics
- Configuration of the Classification Model
- Detection of Leaf disease

DESIGN AND METHODOLOGIES

DATASET COLLECTION:

Data is a crucial part of any Machine Learning System. Datasets from various government websites and kaggle were used to predict the disease leaf. The dataset for the plant disease system consists of 22 plants grown across India. Dataset for plant leaf disease classification consists of images of leaves of 14 plants while excluding healthy leaves, 26 types of images that show a particular disease in a plant. For each plant disease type, there are 1800 images.

DESIGN AND METHODOLOGIES

DATA PREPROCESSING

- Labels of each plant images are then also mapped to a unique the pre-processing step for any machine learning model is of great importance and ideally shapes the performance and results of the models chosen. In this report, the following were the steps that were carried out in order to make sure that the models produced optimal results.

- After reading and resizing the images, we then convert the images into an array form using np .array ()

    - The e value using Label Binarizer()
    - Finally, the plant village dataset is split into two different sets, namely, train and test set with a 75:25 ratio respectively.

DESIGN AND METHODOLOGIES

FEATURE EXTRACTION:

- Feature extraction is a process in machine learning where relevant information is extracted from raw data to create a more meaningful and simplified representation of the data that can be easily processed by a learning algorithm. For the plant leaf disease image dataset, feature extraction involves extracting features such as color, texture, and shape of the leaves. This can be done using image processing techniques such as edge detection, color histograms, and convolutional neural networks (CNNs) to identify and extract these features. These features are then fed into a machine learning algorithm for classification or prediction of the type of plant disease present.

DESIGN AND METHODOLOGIES

MODEL SELECTION AND METRICS:

- It involves selecting the appropriate algorithm and architecture to use in the model, as well as tuning the hyperparameters to achieve the best performance. For plant leaf disease detection using image dataset, deep learning architectures such as convolutional neural networks (CNNs) are commonly used. CNNs are particularly suited to image data as they can automatically extract relevant features from images without the need for manual feature engineering. Transfer learning, where a pre-trained CNN model is fine-tuned for the specific task of plant leaf disease detection, can also be used to improve performance with limited data.

DESIGN AND METHODOLOGIES

CONFIGURATION OF THE CLASSIFICATION MODEL:

- The architecture of CNN used for plant disease detection in this project was as follows, the first block contains a Convolutional layer with 32 filters of size 3 x 3 and the activation function used was the ReLU activation function. We then follow the operation by performing batch normalization, and choosing the Max Pooling layer with a pool size of and adding a dropout layer with 25% drop- out. Batch normalization was performed in order to speed up the convergence of the neural network, it is generally applied after each individual layer so that the out- put of the previous layer can be normalized allowing for each individual layer present in the network to perform learning independent. Dropout layer is a technique used to prevent the model from overfitting by randomly switching off some sections of the neurons. When some sections of the neurons are switched off the incoming as well as the outgoing connections from the neurons are also switched off and this results in the betterment of the model in learning and allows for the model to not generalize to the test dataset. We used the pre trained model for leaf disease prediction

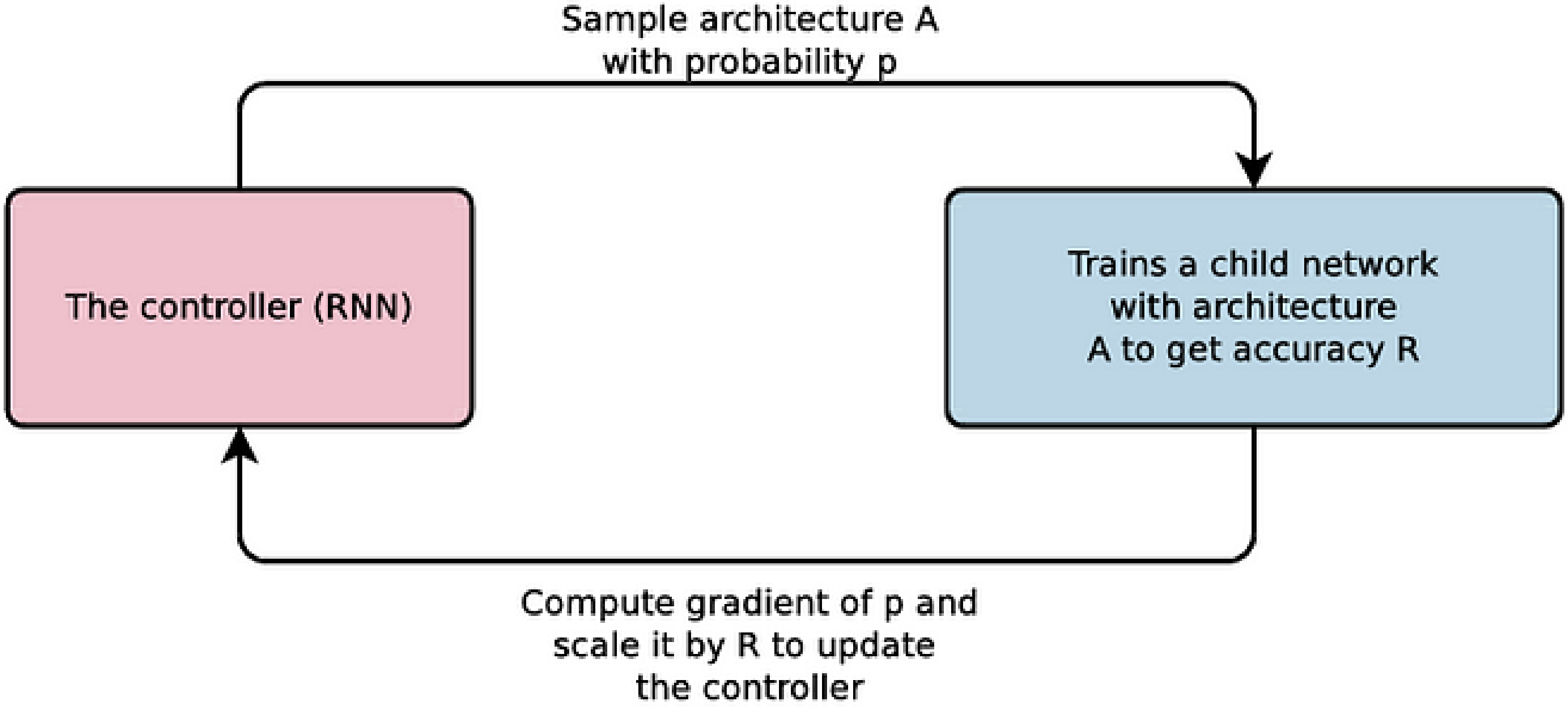
DESIGN AND METHODOLOGIES

DETECTION OF LEAF DISEASE:

- Pre-trained CNN models can be used to detect plant leaf diseases by extracting relevant features from the images and classifying the type of disease present. This approach can improve the accuracy of the predictions and reduce the need for manual feature engineering. However, it requires a large amount of data to fine-tune the pre-trained model for the specific task. Use the pre trained CNN model to make predictions on new plant leaf images to detect the type of disease present.

EFFICIENTNETV2 ALGORITHM:

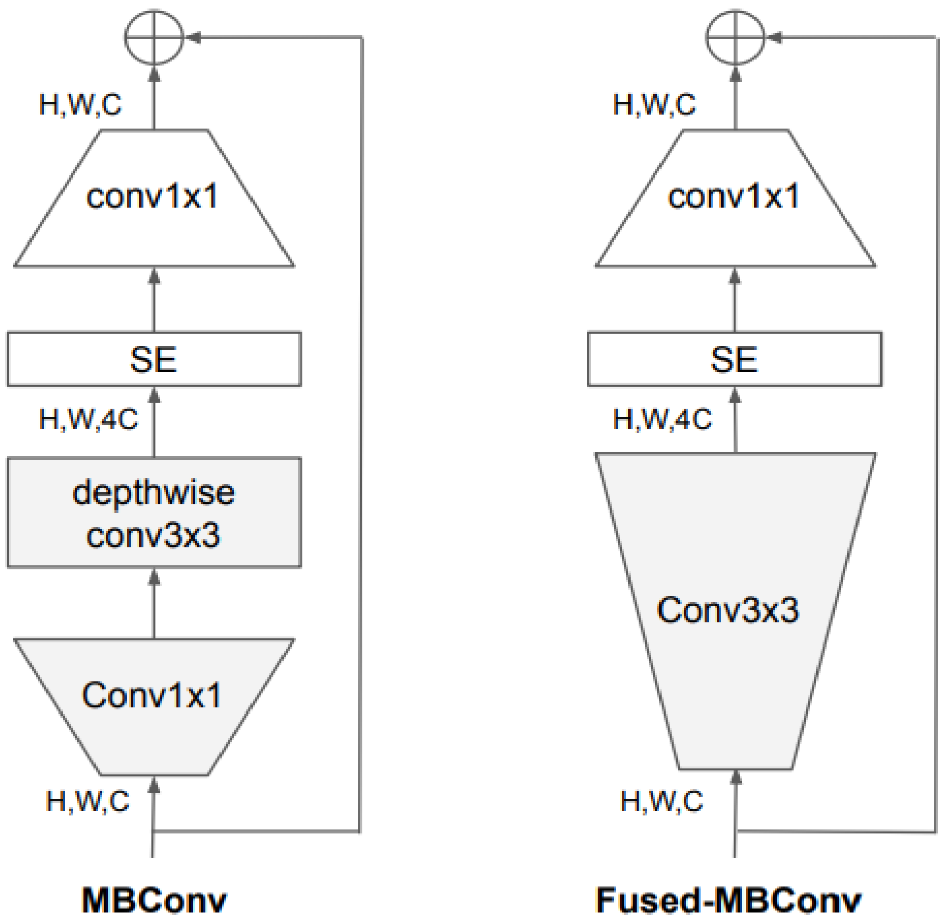
EfficientNets are currently one of the most powerful convolutional neural network (CNN) models. With the rise of Vision Transformers, which achieved even higher accuracies than EfficientNets, the question arose whether CNNs are now dying. EfficientNetV2 proves this wrong by not just improving accuracies but by also reducing training time and latency. The EfficientNet models are designed using neural architecture search. The first neural architecture search was proposed in the paper in 2016 — 'Neural Architecture Search with Reinforcement Learning'. The idea is to use a controller (a network such as an RNN) and sample network architectures from a search space with probability 'p'. This architecture is then evaluated by first training the network, and then validating it on a test set to get the accuracy 'R'. The gradient of 'p' is calculated and scaled by the accuracy 'R'. The result (reward) is fed to the controller RNN. The controller acts as the agent, the training and testing of the network act as the environment, and the result acts as the reward. This is the common Reinforcement learning (RL).

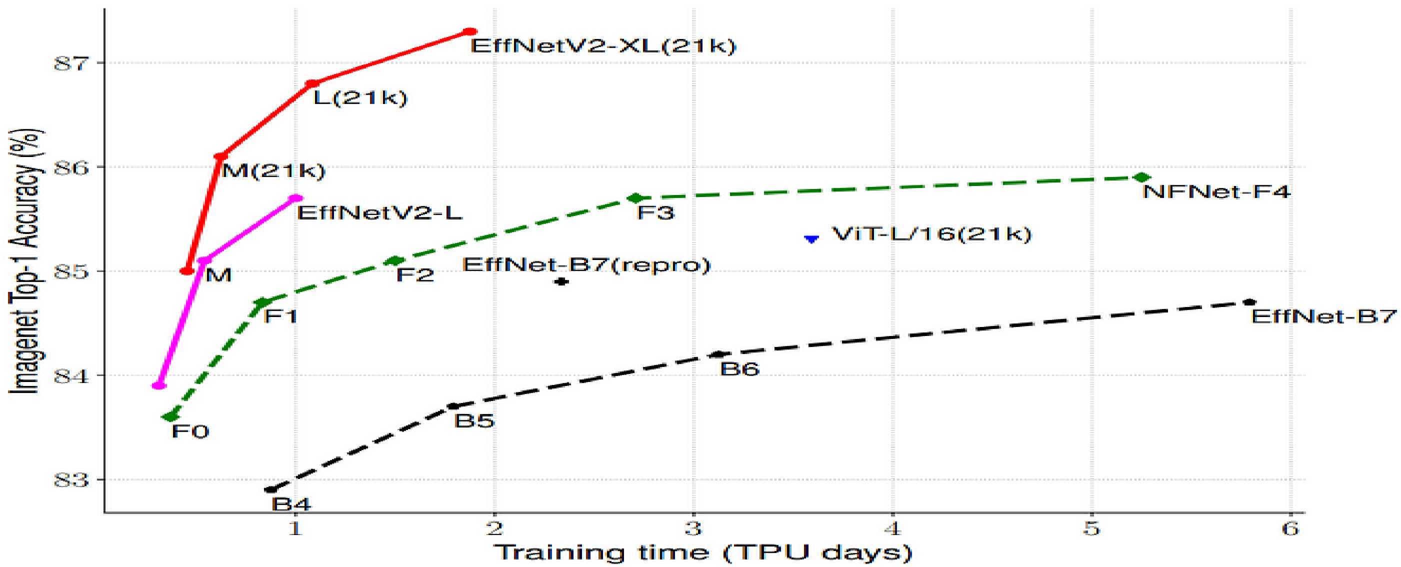

Sample architecture A
with probability p
The controller (RNN)
Trains a child network
with architecture
A to get accuracy R
Compute gradient of p and
scale it by R to update
the controller

EfficientNetV2

- EfficientNetV2 goes one step further than EfficientNet to increase training speed and parameter efficiency. This network is generated by using a combination of scaling (width, depth, resolution) and neural architecture search. The main goal is to optimize training speed and parameter efficiency. Also, this time the search space also included new convolutional blocks such as Fused-MBConv. In the end, the authors obtained the EfficientNetV2 architecture which is much faster than previous and newer state-of-the-art models and is much smaller (up to 6.8x times).

- clearly shows that The EfficientnetV2 has 24 million parameters, while a Vision Transformer (ViT) has 86 million parameters. The V2 version also has nearly half the parameters of the original EfficientNet. While it does reduce the parameter size significantly, it maintains similar or higher accuracies than the other models on the ImageNet dataset.


H,W,C
conv1x1
SE
H,W,4C
depthwise
conv3x3
Conv1x1
H,W,C
MBConv
H,W,C
conv1x1
SE
H,W,4C
Conv3x3
H,W,C
Fused-MBConv



(a) Training efficiency.

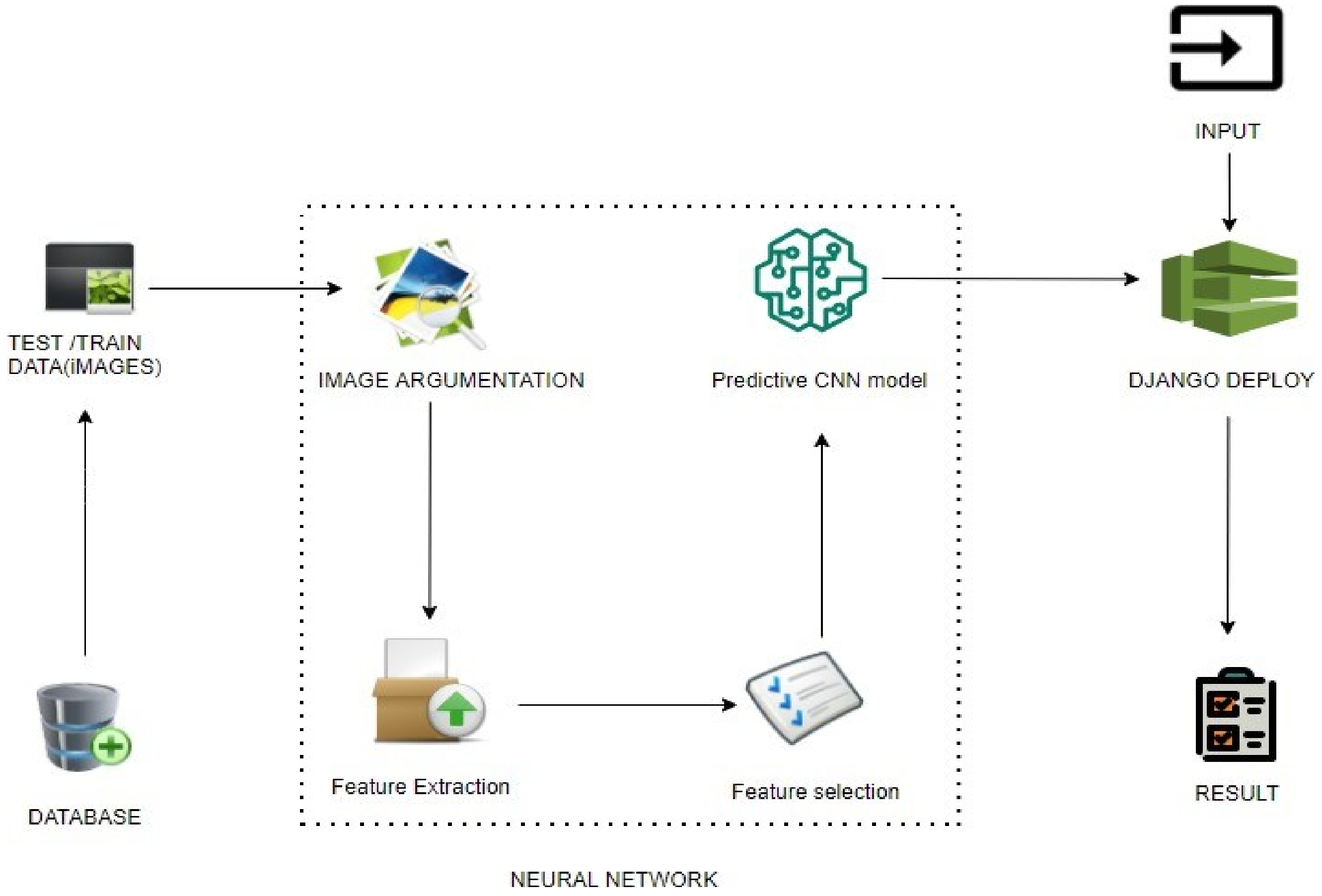
	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

IMPLEMENTATION

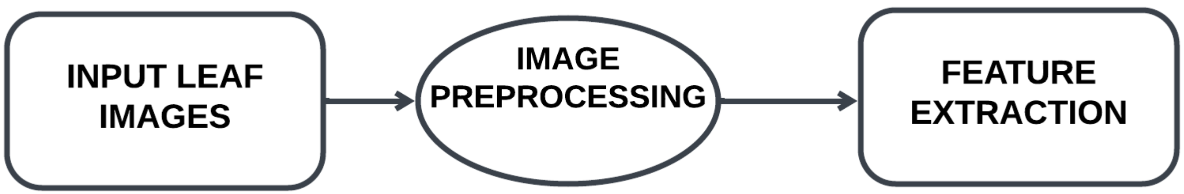
- ARCHITECTURE DIAGRAM
- DATA FLOW DIAGRAM
- ER DIAGRAM
- SEQUENCE DIAGRAM
- COLLABORATION DIAGRAM

ARCHITECTURE DIAGRAM



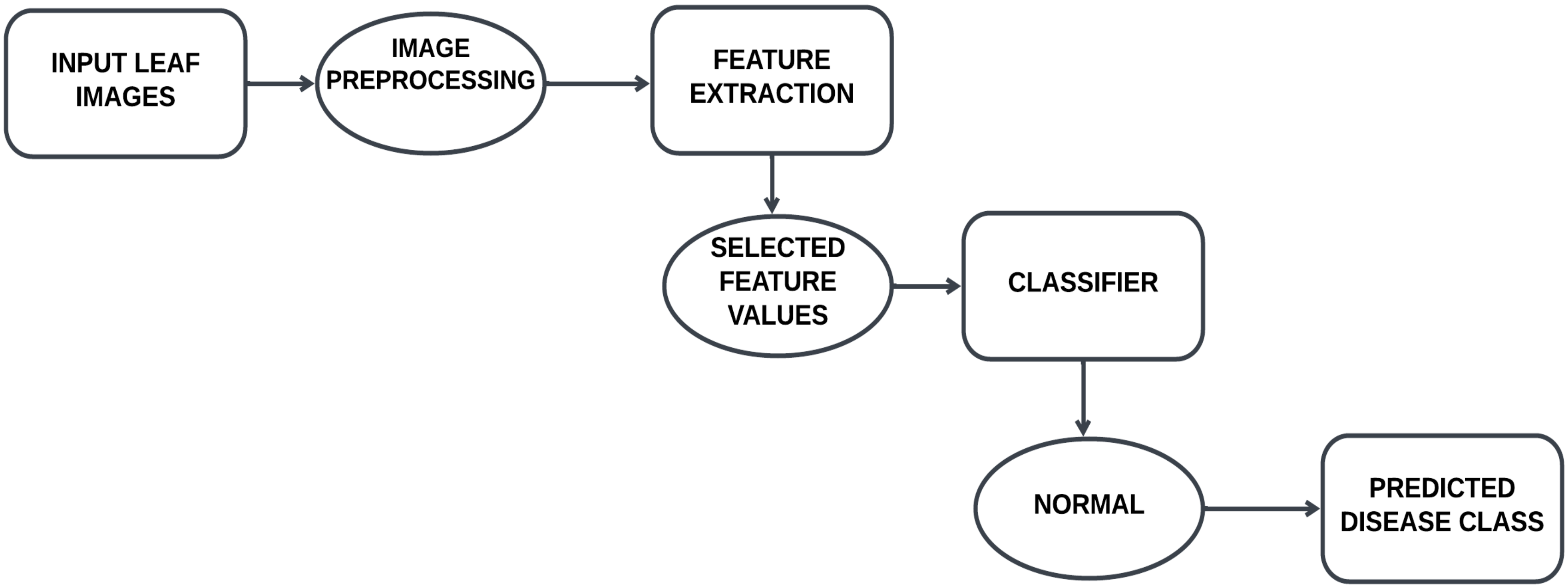
DATA FLOW DIAGRAM

LEVEL 0:

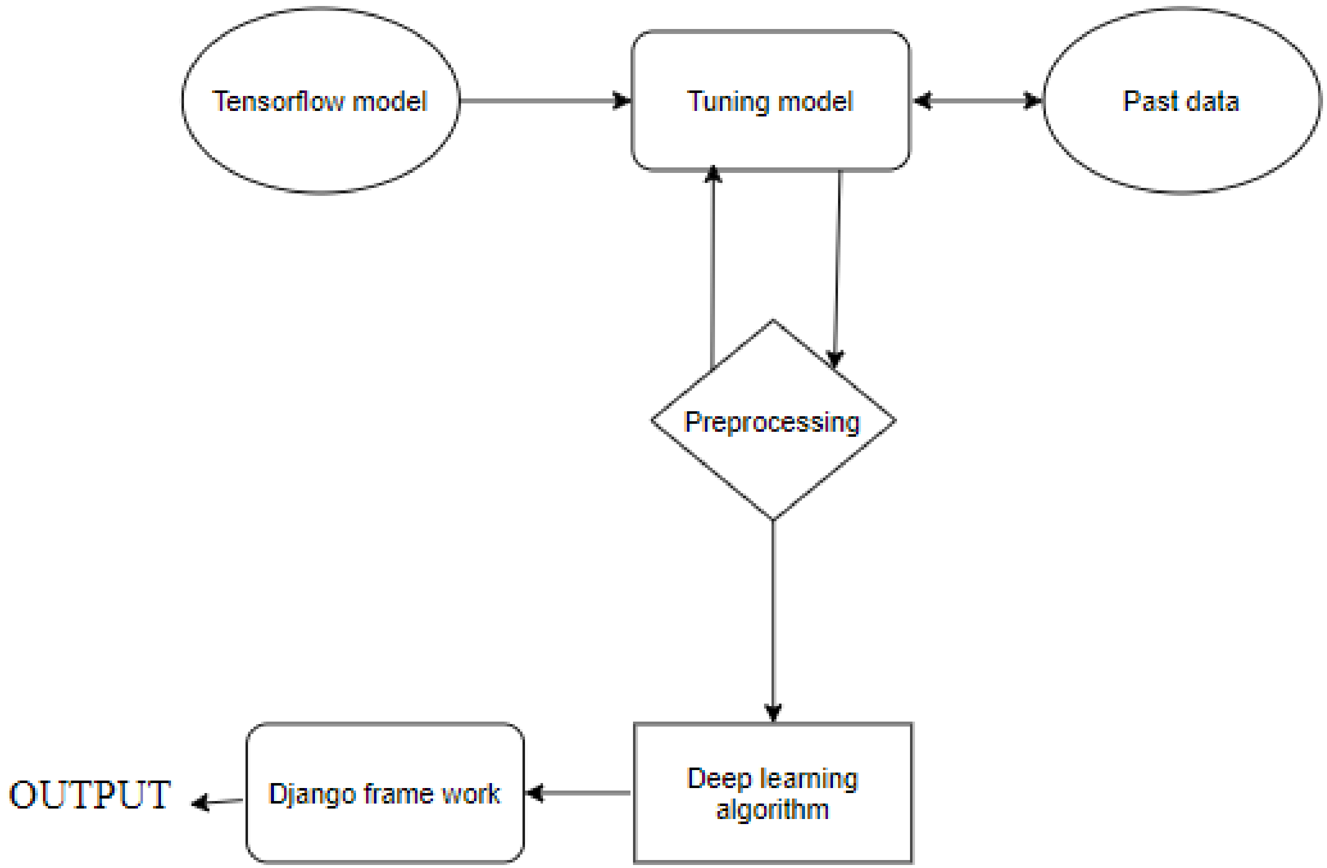


LEVEL 1:

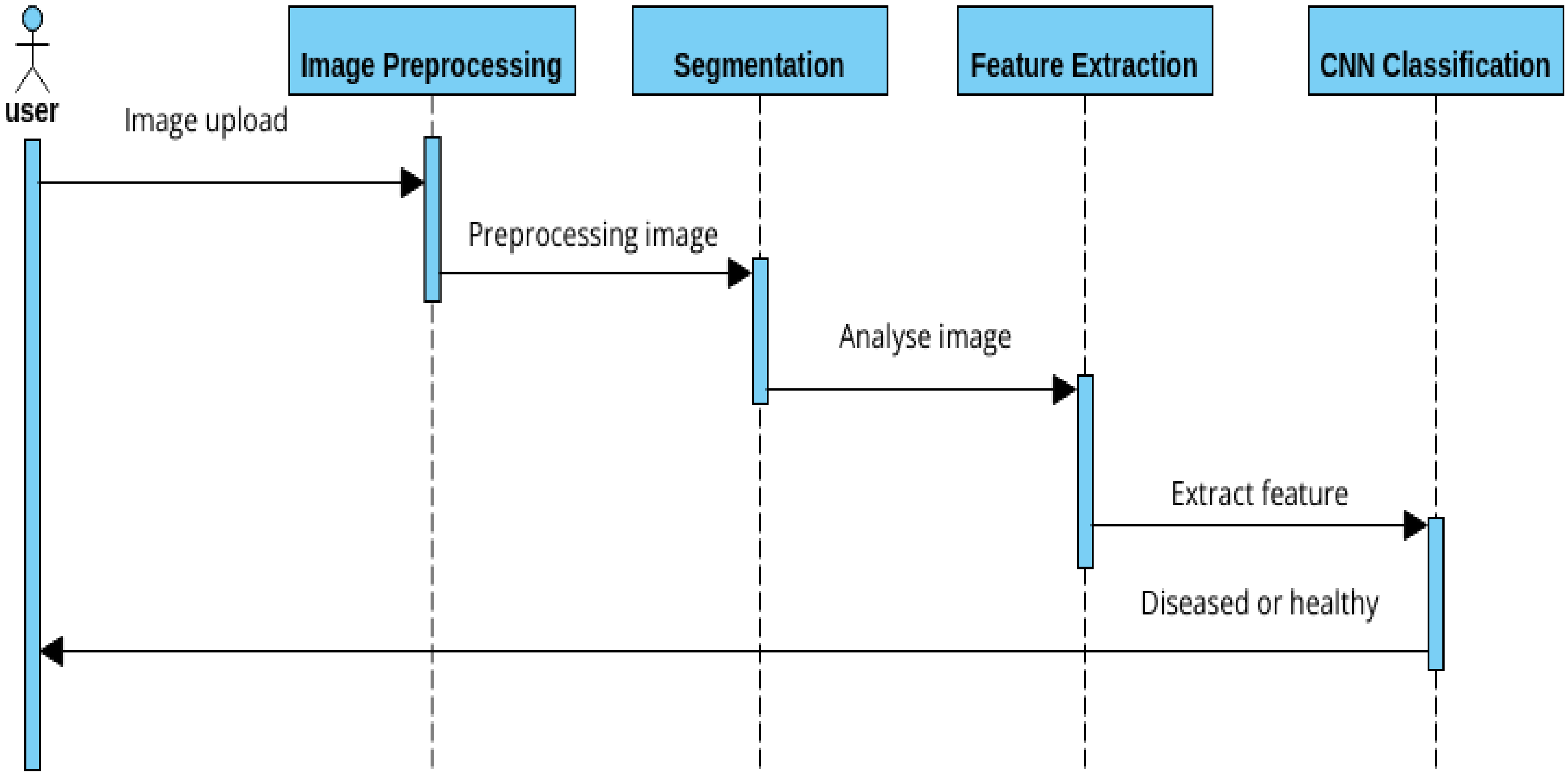
DATA FLOW DIAGRAM



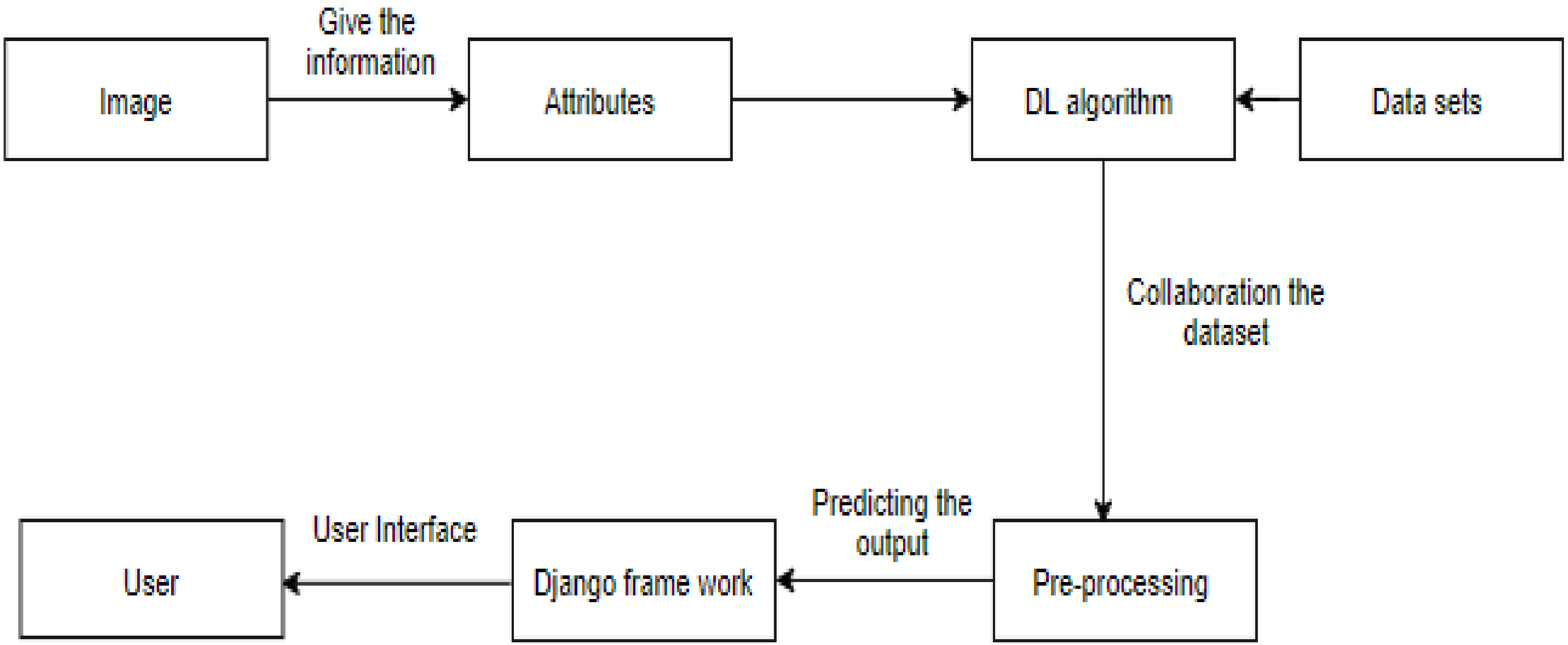
ER DIAGRAM


Tensorflow model
Tuning model
Past data
Preprocessing
Deep learning algorithm
Django frame work
OUTPUT

SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



TESTING

- UNIT TESTING
- INTEGRATION TESTING
- FUNCTIONAL TESTING
- WHITE BOX TESTING
- BLACK BOX TESTING

UNIT TESTING

```
import unittest
import numpy as np
from your_module import DiseaseDetectorCNN  # Import your CNN model class or function

class TestPlantDiseaseDetection(unittest.TestCase):

    def setUp(self):
        # Initialize your CNN model
        self.model = DiseaseDetectorCNN()

    def tearDown(self):
        # Clean up resources if needed
        pass

    def test_disease_prediction(self):
        # Test case for disease prediction

        # Generate a sample input image (you may use actual plant images)
        # Here, we create a dummy image with shape (height, width, channels)
        input_image = np.random.rand(224, 224, 3)  # Assuming input shape of your CNN model

        predicted_disease = self.model.predict(input_image)

         expected_classes = ["disease1", "disease2", "disease3“]
```

INTEGRATION TESTING

```
import unittest
import numpy as np
from your_module import DiseaseDetectorSystem  # Import your integrated system class or function

class TestIntegratedPlantDiseaseDetection(unittest.TestCase):

    def setUp(self):
        # Initialize your integrated system
        self.system = DiseaseDetectorSystem()

    def tearDown(self):
        # Clean up resources if needed
        pass

    def test_disease_detection(self):
        # Test case for disease detection

        # Generate a sample input image (you may use actual plant images)
        # Here, we create a dummy image with shape (height, width, channels)
        input_image = np.random.rand(224, 224, 3)  # Assuming input shape of your CNN model

        # Perform disease detection using the integrated system
        detection_result = self.system.detect_disease(input_image)
```

INTEGRATION TESTING

```
# Ensure that the detection result is valid and includes predicted disease
        self.assertTrue("disease" in detection_result, "Detection result should include predicted 
disease")

        # Ensure that the confidence score is within a reasonable range
        confidence_score = detection_result["confidence"]
        self.assertTrue(0 <= confidence_score <= 1, "Confidence score should be between 0 and 1")

        # Ensure that additional information is provided, if available
        if "additional_info" in detection_result:
            additional_info = detection_result["additional_info"]
            self.assertIsInstance(additional_info, dict, "Additional info should be a dictionary")
            # Add more assertions for specific additional information if needed
```

FUNCTIONAL TESTING

```
import unittest
import numpy as np
from your_module import DiseaseDetectorSystem

class TestFunctionalPlantDiseaseDetection(unittest.TestCase):

    def setUp(self):
        # Initialize your integrated system
        self.system = DiseaseDetectorSystem()

    def tearDown(self):
        # Clean up resources if needed
        pass

    def test_detection_with_real_image(self):
        # Test case for disease detection with a real image

        input_image = load_image("path_to_image")

        # Perform disease detection using the integrated system
        detection_result = self.system.detect_disease(input_image)

        self.assertTrue("disease" in detection_result, "Detection result should include predicted disease")

        # Ensure that the confidence score is within a reasonable range
        confidence_score = detection_result["confidence"]
        self.assertTrue(0 <= confidence_score <= 1, "Confidence score should be between 0 and 1")
```

FUNCTIONAL TESTING

```
# Ensure that additional information is provided, if available
    if "additional_info" in detection_result:
        additional_info = detection_result["additional_info"]
        self.assertIsInstance(additional_info, dict, "Additional info should be a dictionary")
        # Add more assertions for specific additional information if needed

  def test_detection_with_augmented_image(self):
    # Test case for disease detection with an augmented image

    # Generate a synthetic augmented image
    augmented_image = generate_augmented_image()

    # Perform disease detection using the integrated system
    detection_result = self.system.detect_disease(augmented_image)

    # Ensure that the detection result is valid and includes predicted disease
    self.assertTrue("disease" in detection_result, "Detection result should include predicted disease")

    # Ensure that the confidence score is within a reasonable range
    confidence_score = detection_result["confidence"]
    self.assertTrue(0 <= confidence_score <= 1, "Confidence score should be between 0 and 1")

    # Ensure that additional information is provided, if available
    if "additional_info" in detection_result:
        additional_info = detection_result["additional_info"]
        self.assertIsInstance(additional_info, dict, "Additional info should be a dictionary")
        # Add more assertions for specific additional information if needed
```

WHITE BOX TESTING

```
import unittest
import numpy as np
from your_module import CNNModel  # Import your CNN model class or function
class TestWhiteBoxPlantDiseaseDetection(unittest.TestCase):
    def setUp(self):
        self.model = CNNModel()
   def tearDown(self):
        pass
   def test_model_architecture(self):
        self.assertEqual(len(self.model.layers), 10)
        expected_layer_config = [...]  # Define expected layer configurations
        for i, layer in enumerate(self.model.layers):
            self.assertDictEqual(layer.get_config(), expected_layer_config[i])
   def test_model_training(self):
        X_train = np.random.rand(100, 224, 224, 3)
        y_train = np.random.randint(0, 3, size=(100,))
        history = self.model.train(X_train, y_train, epochs=5, batch_size=32)
        self.assertLess(history.history['loss'][-1], history.history['loss'][0])
        self.assertGreater(history.history['accuracy'][-1], history.history['accuracy'][0])
   def test_model_inference(self):
        input_image = np.random.rand(224, 224, 3)
        predicted_class = self.model.predict(input_image)
        self.assertTrue(0 <= predicted_class <= 2)
if __name__ == '__main__':
    unittest.main()
```

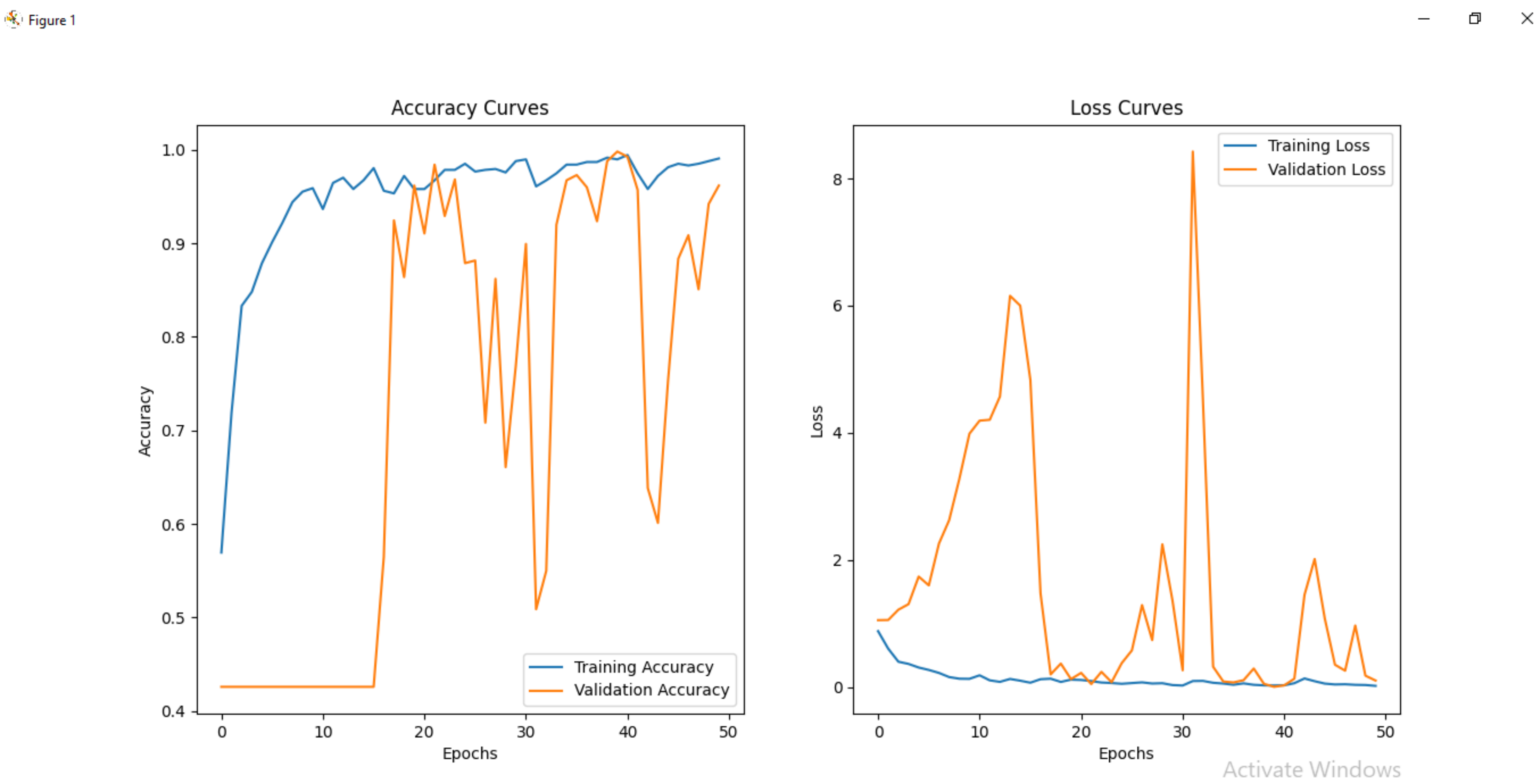
BLACK BOX TESTING

```
import unittest
import numpy as np
from your_module import DiseaseDetectorSystem  # Import your integrated system class or function
class TestBlackBoxPlantDiseaseDetection(unittest.TestCase):
  def setUp(self):
      self.system = DiseaseDetectorSystem()
  def tearDown(self):
      pass
  def test_detection_with_real_images(self):
      input_images = [load_image("path_to_image1"), load_image("path_to_image2"), ...]
      for input_image in input_images:
          detection_result = self.system.detect_disease(input_image)
          self.assertTrue("disease" in detection_result)
          self.assertTrue(0 <= detection_result["confidence"] <= 1)
          if "additional_info" in detection_result:
              self.assertIsInstance(detection_result["additional_info"], dict)
   def test_detection_with_augmented_images(self):
      augmented_images = [generate_augmented_image() for _ in range(10)]
      for augmented_image in augmented_images:
          detection_result = self.system.detect_disease(augmented_image)
          self.assertTrue("disease" in detection_result)
          self.assertTrue(0 <= detection_result["confidence"] <= 1)
          if "additional_info" in detection_result:
              self.assertIsInstance(detection_result["additional_info"], dict)
if __name__ == '__main__':
   unittest.main()
```

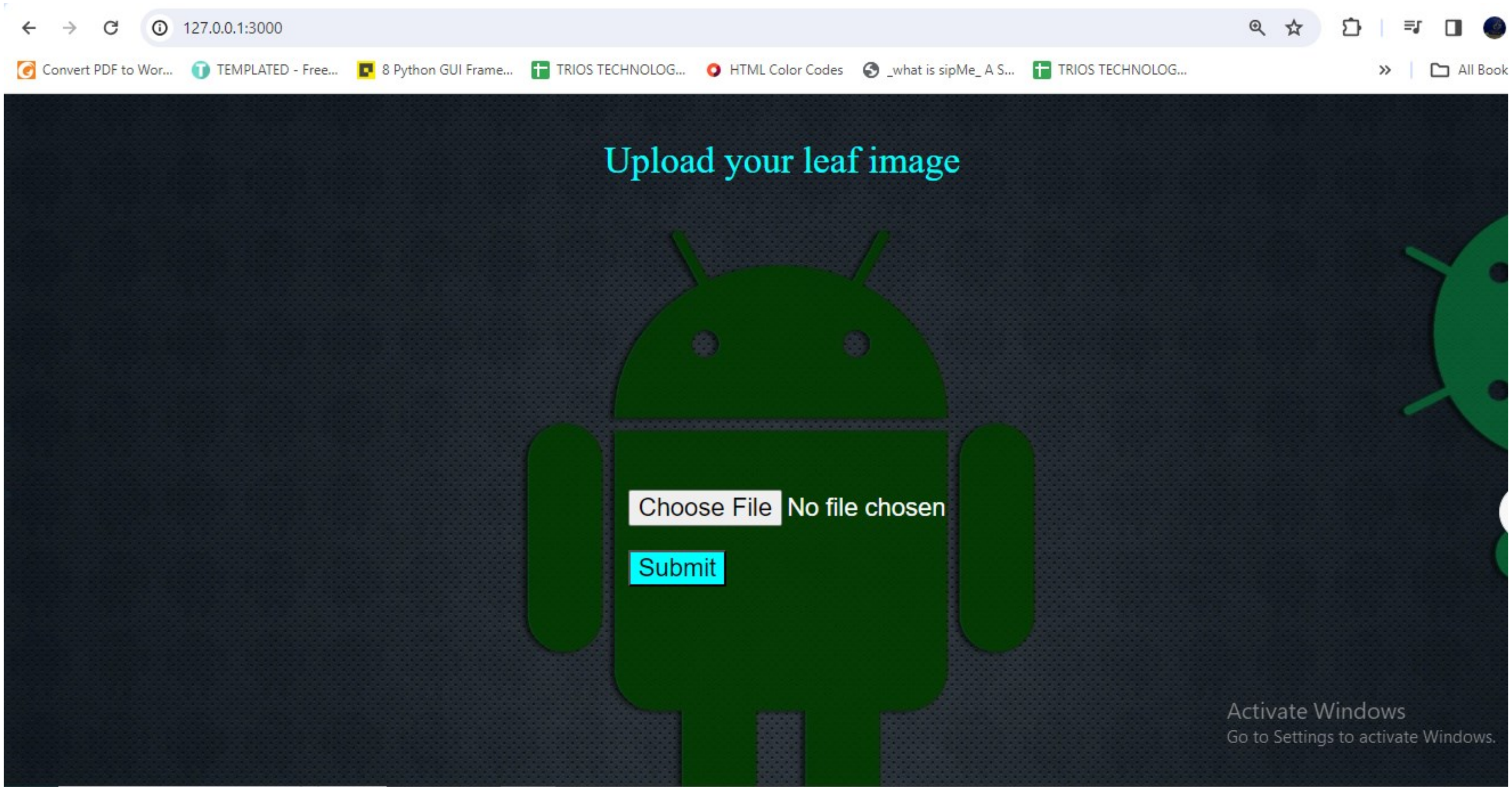
INPUT AND OUTPUT

```
Epoch 1/50
56/56 [==============================] - 269s 5s/step - loss: 0.4251 - accuracy: 0.8467 -
0.7580
Epoch 2/50
56/56 [==============================] - 266s 5s/step - loss: 0.4346 - accuracy: 0.8484 -
0.7522
Epoch 3/50
56/56 [==============================] - 229s 4s/step - loss: 0.4156 - accuracy: 0.8533 -
0.7697
Epoch 4/50
56/56 [==============================] - 219s 4s/step - loss: 0.4064 - accuracy: 0.8604 -
0.7697
Epoch 5/50
56/56 [==============================] - 237s 4s/step - loss: 0.4086 - accuracy: 0.8566 -
0.7609
Epoch 6/50
56/56 [==============================] - 245s 4s/step - loss: 0.4083 - accuracy: 0.8588 -
0.7434
Epoch 7/50
26/56 [============>.................] - ETA: 1:43 - loss: 0.3859 - accuracy: 0.8575
```

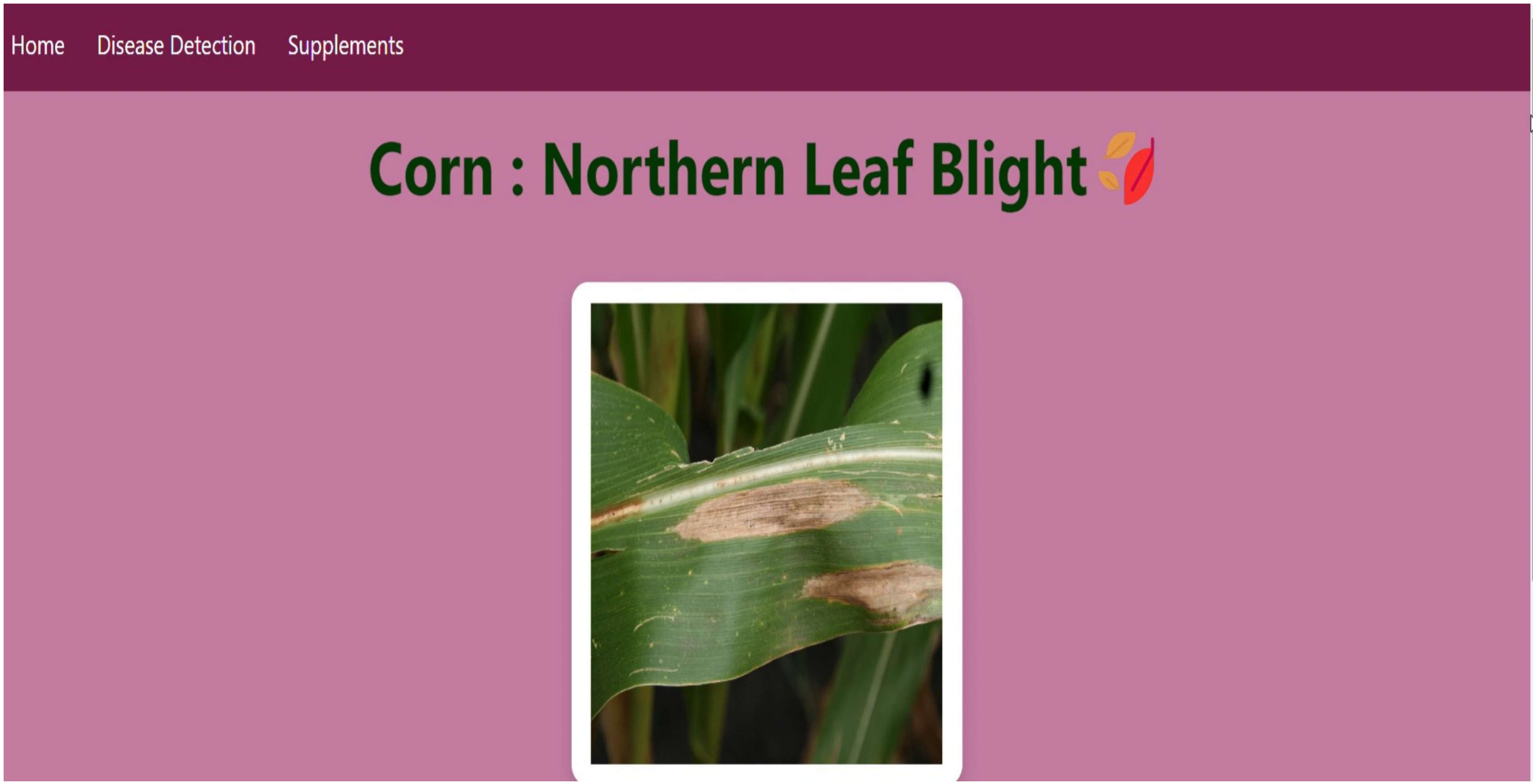
INPUT AND OUTPUT



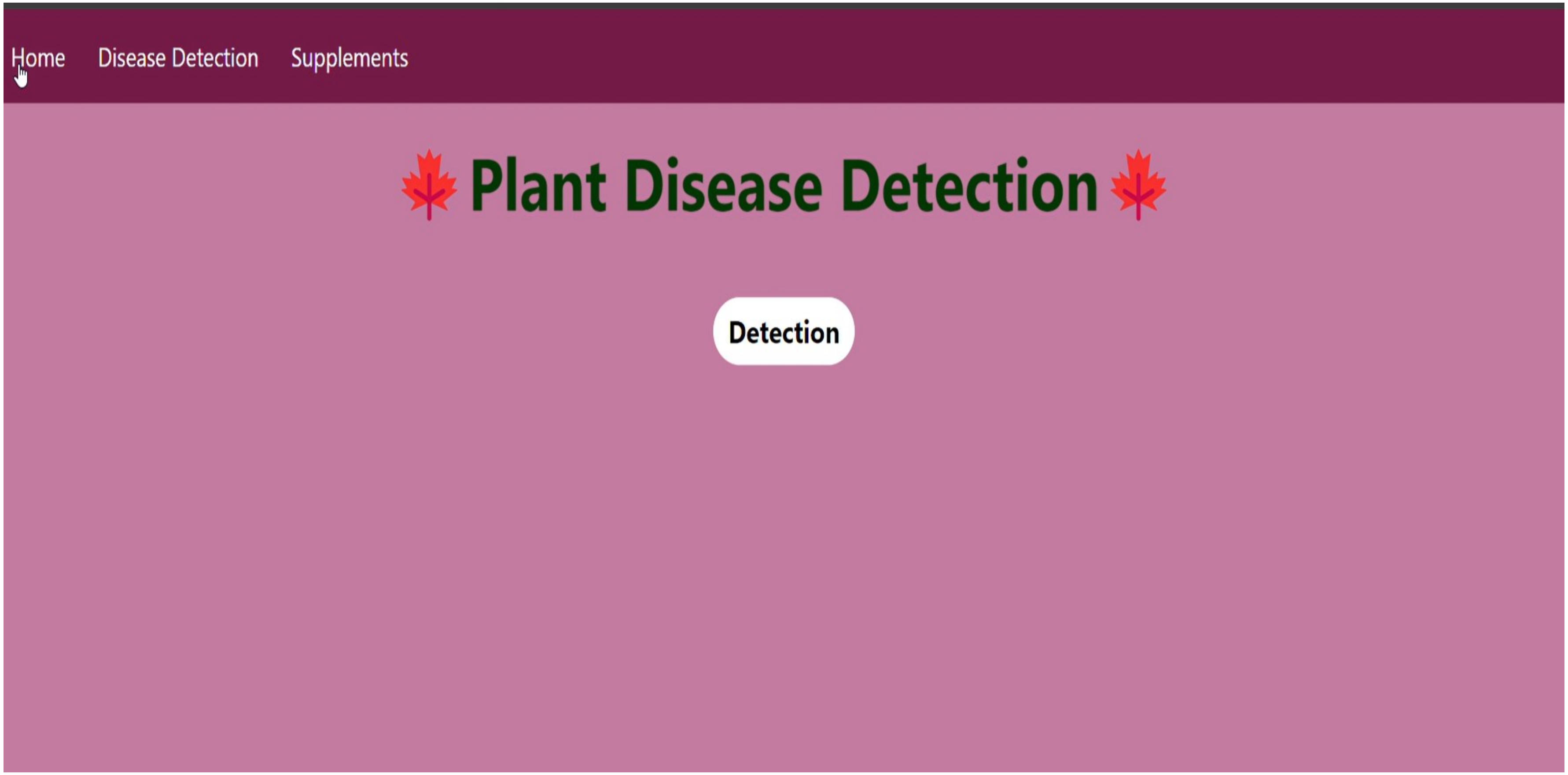
INPUT AND OUTPUT



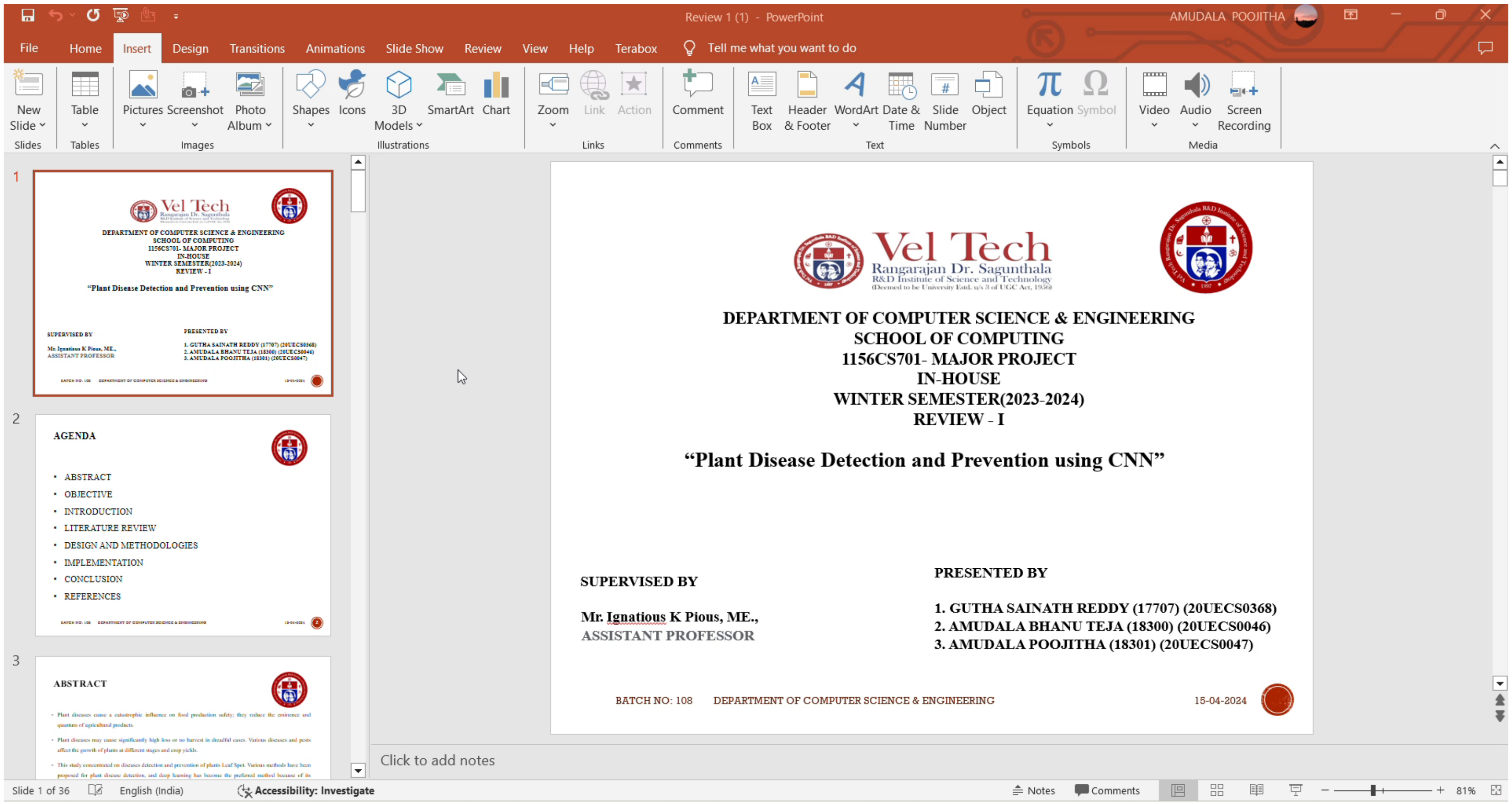
INPUT AND OUTPUT



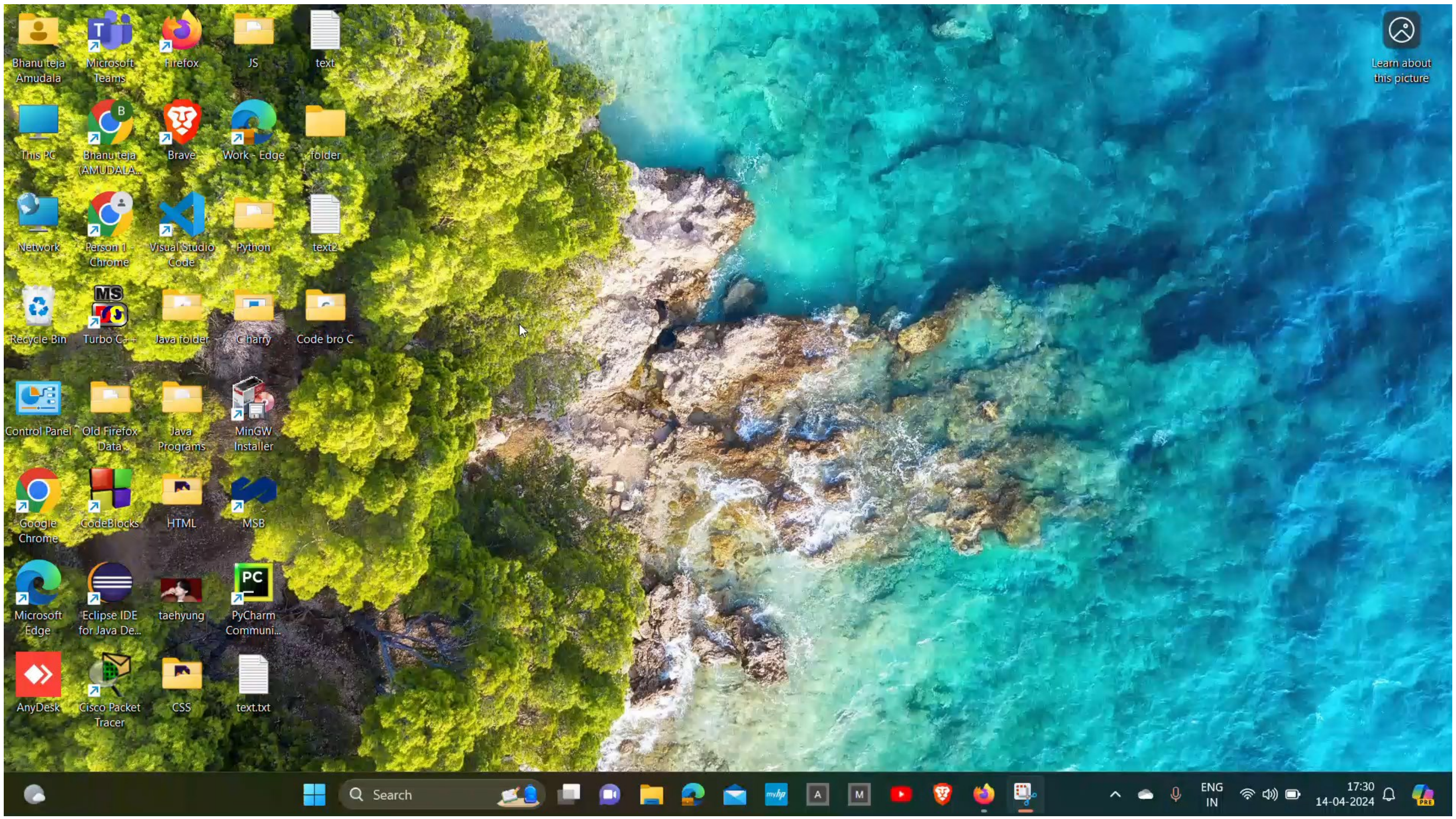
INPUT AND OUTPUT



DEMO VIDEO-1


Review 1 (1) - PowerPoint
AMUDALA POOJITHA
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Terabox
Tell me what you want to do
New Slide
Table
Pictures
Screenshot
Photo Album
Shapes
Icons
3D Models
SmartArt
Chart
Zoom
Link
Action
Comment
Text Box
Header & Footer
WordArt
Date & Time
Slide Number
Object
Equation
Symbol
Video
Audio
Screen Recording
Slides
Tables
Images
Illustrations
Links
Comments
Text
Symbols
Media
AGENDA
ABSTRACT
OBJECTIVE
INTRODUCTION
LITERATURE REVIEW
DESIGN AND METHODOLOGIES
IMPLEMENTATION
CONCLUSION
REFERENCES
ABSTRACT
Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING
1156CS701- MAJOR PROJECT
IN-HOUSE
WINTER SEMESTER(2023-2024)
REVIEW - I
“Plant Disease Detection and Prevention using CNN”
SUPERVISED BY
Mr. Ignatious K Pious, ME.,
ASSISTANT PROFESSOR
PRESENTED BY
1. GUTHA SAINATH REDDY (17707) (20UECS0368)
2. AMUDALA BHANU TEJA (18300) (20UECS0046)
3. AMUDALA POOJITHA (18301) (20UECS0047)
BATCH NO: 108
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
15-04-2024
Click to add notes
Slide 1 of 36
English (India)
Accessibility: Investigate
Notes
Comments
81%

DEMO VIDEO-2


Bhanu teja Amudala
Microsoft Teams
Firefox
JS
text
This PC
Bhanu teja (AMUDALA...
Brave
Work - Edge
folder
Network
Person 1 - Chrome
Visual Studio Code
Python
text2
Recycle Bin
Turbo C++
Java folder
Charry
Code bro C
Control Panel
Old Firefox Data
Java Programs
MinGW Installer
Google Chrome
CodeBlocks
HTML
MSB
Microsoft Edge
Eclipse IDE for Java De...
taehyung
PyCharm Communi...
AnyDesk
Cisco Packet Tracer
CSS
text.txt
Learn about this picture
Search
ENG IN
17:30
14-04-2024

CONCLUSION

- An efficient plant leaf disease detection approach is essential to detect plant diseases in real-time. In this regard, the plant leaf disease detection approach is proposed, where the cardamom plant leaf dataset was collected from a farmland with a complex background.

- Segmenting and detecting diseases in real-time images is a challenging task, as the images are associated with other factors such as the background of the image, environmental factors such as lighting, and angle of the capturing conditions.

- In the proposed method, the U2 -Net architecture is employed to remove the complex background, which produces results without deteriorating the quality of the original image.

- For classification, in this work, CNN, EfficientNet, and EfficientNetV2 models were trained instead of using the pre-trained weights for EfficientNet, and EfficientNetV2.

WEB REFERENCES LINK

- https://www.researchgate.net/publication/349162145_PLANT_DISEASE_DETECTION_USING_CONVOLUTIONAL_NEURAL_NETWORK
- https://www.itmconferences.org/articles/itmconf/pdf/2022/04/itmconf_icacc2022_03049.pdf
- https://www.semanticscholar.org/paper/Plant-Disease-Detection-using-Convolutional-Neural-Srivastava-Mishra/eaa0325ba66f1aa8ffdc6025aa21d3fbad928425
- https://www.academia.edu/65147374/Plant_Disease_Detection_Using_Convolutional_Neural_Network
- https://link.springer.com/article/10.1007/s41870-020-00437-5

REFERENCES

[1] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans leaf diseases classification using MobileNet models," IEEE Access, vol. 10, pp. 9471–9482, 2022. PeerComput.Sci.,vol.7,p.e352,Mar.2021

[2] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning are view,"IEEEAccess,vol.9,pp.56683–56698,2021.

[3] D. Wang, J. Wang, Z. Ren, and W. Li, "DHBP: A dual-stream hierarchical bilinear pooling model for plant disease multi-task classification," Comput. Electron. Agricult., vol.195,Apr.2022,Art.no.106788.

[4] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning,"IEEEAccess,vol.9,pp.140565 140580,2021.

[5]L.C.Ngugi,M.Abdelwahab,andM.Abo-Zahhad,"Tomatoleaf segmentation algorithms for mobile phone applications using deep learning," Comput. Electron. Agricult., vol. 178, Nov. 2020, Art. no. 105788

REFERENCES

[6] AI Challenger.(2018). AI Challenger 2018 Datasets. Accessed: Apr. 1, 2022. [Online].

[7] D.O.Oyewola,E.G.Dada,S. Misra, and R. Damaševičius,"Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing, "PeerJComput. Sci .,vol . 7, p.e352 ,Mar.2021

[8] Makerere AI Lab and AIR Lab National Crops Resources Research Institute (NaCRRI) .(2020).

iBean . Accessed :Mar.31,2022

[9] M.Adi, A. K. Singh, H. Reddy A. Y. Kumar, V. R. Challa, P. Rana, and U. Mittal, "An overview on plant disease detection algorithm using deep learning," in Proc.2nd Int. Conf.Intell.Eng.Manage .(ICIEM) ,Apr.2021,pp.305–309

[10] C.K.Sunil,C.D.Jaidhar,and N. Patil, "Cardamom plant disease detec[1]tion approach using

EfficientNetV2," IEEE Access, vol. 10, pp. 789–804, 2022

THANK YOU