
PLANT DISEASE DETECTION

PROBLEM STATEMENT

Build a machine learning model to detect the disease of plant with the picture of diseased plant leaves.

EDA

EXPLORATORY DATA ANALYSIS

BY CONDUCTING EDA AND ANALYZING THE DISTRIBUTION OF CLASSES IN THE DATASET, WE GAINED INSIGHTS INTO THE DATASET'S STRUCTURE AND IDENTIFIED ANY POTENTIAL CHALLENGES OR IMBALANCES. THIS UNDERSTANDING SERVES AS A FOUNDATION FOR THE SUBSEQUENT STAGES OF THE PROJECT, INCLUDING DATA PREPROCESSING AND MODEL TRAINING.

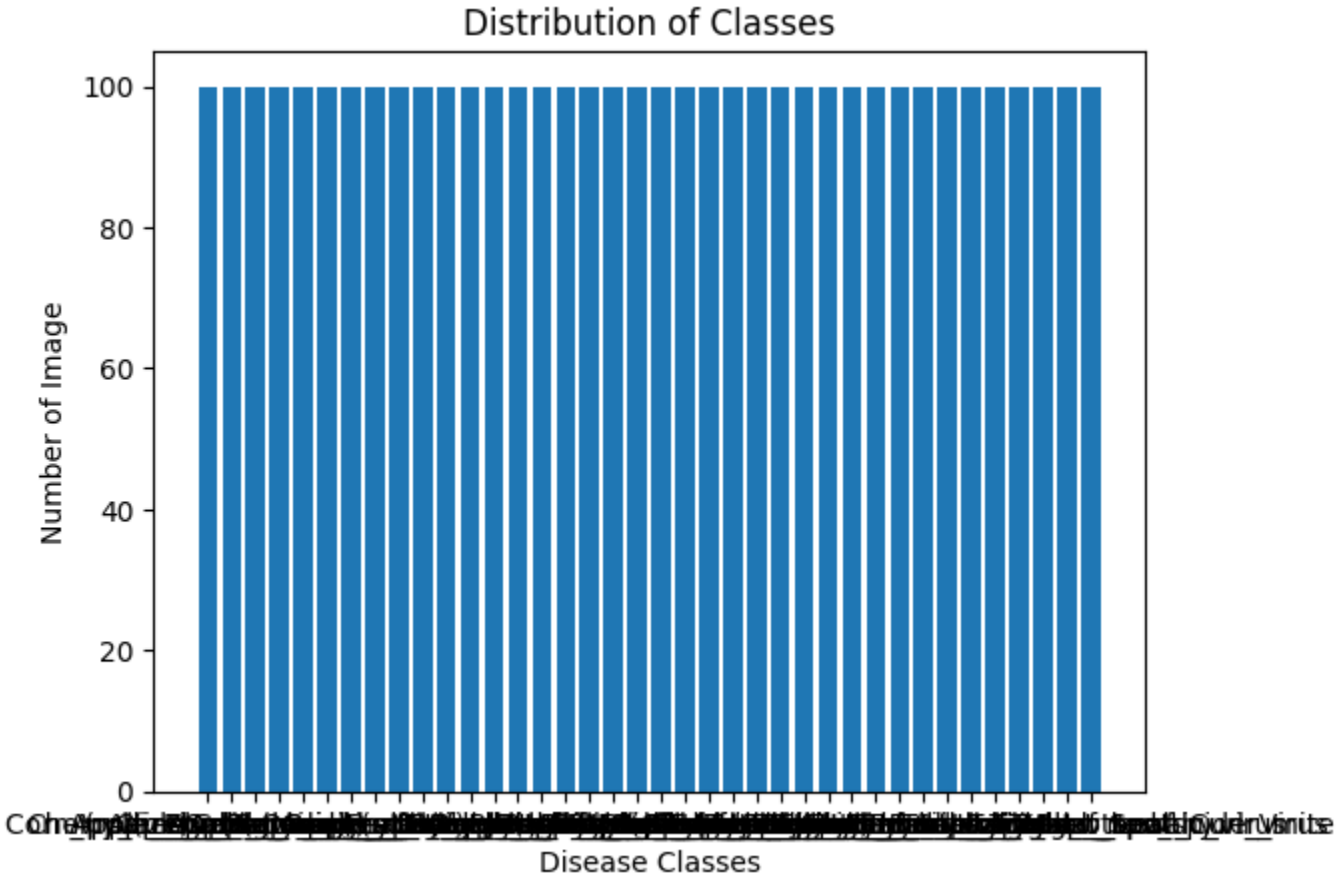
SOURCE

The dataset was sourced from Kaggle.

DATA OVERVIEW

- It consists of a total of images 2,569 and is categorized into 38 different classes based on the types of plant diseases.
- We visualized the distribution of images among different disease classes using bar charts.
- This analysis helps identify any potential class imbalances that might affect model training and performance.
- Each bar represents a disease class, and the height of the bar indicates the number of images in that class.

BAR CHART


Distribution of Classes
Number of Image
100
80
60
40
20
0
Disease Classes

CODE IMPLEMENTATION.

IMPORT LIBRARIES

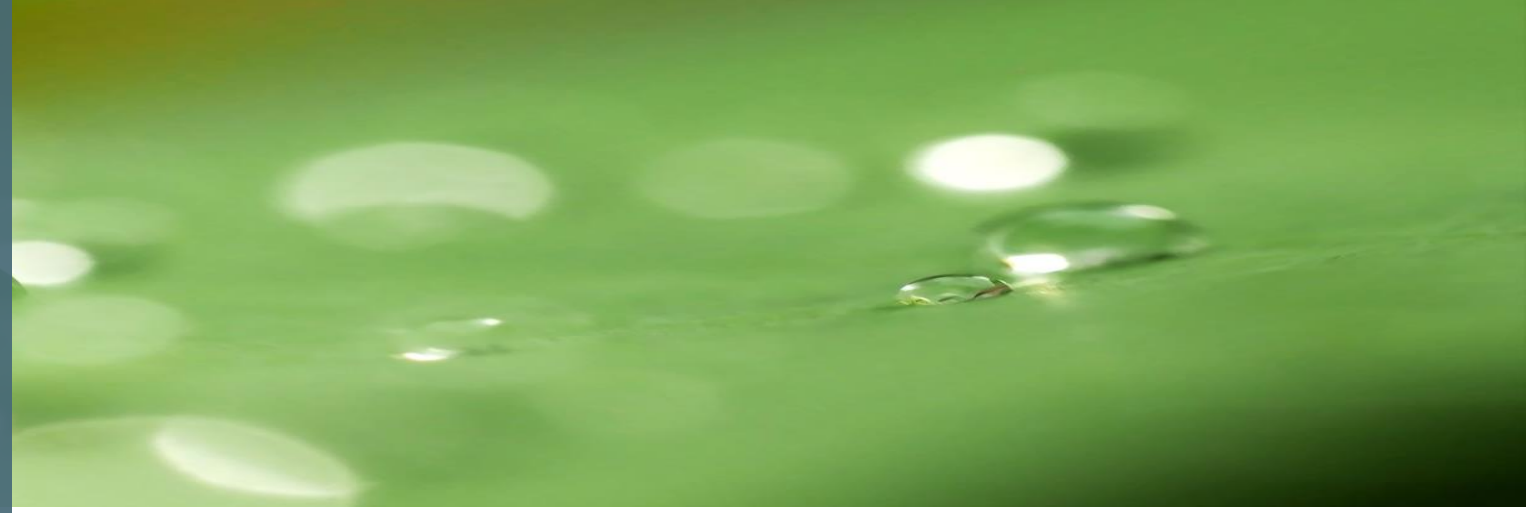
DATA PRE-
PROCESSING

MODEL TRAINING
AND EVALUATION

RESULTS AND
INTERPRETATION

PREDICTING PLANT
DISEASES

EXAMPLE
PREDICTIONS



IMPORT LIBRARIES

```
import numpy as np
import pickle
import cv2
import os
import matplotlib.pyplot as plt
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
# ... (import other libraries)
```

DATA PROCESSING

- Defining the image size and number of images

```
DEFAULT_IMAGE_SIZE = tuple((256,
256))
N_IMAGES = 100
```

- Specifying the dataset paths

```
root_dir = './New Plant Diseases Dataset(Augmented)'
train_dir = os.path.join(root_dir, 'train')
val_dir = os.path.join(root_dir, ‘valid')
```

- Function to convert images to arrays

```
def convert_image_to_array(image_dir):
  # ...
```

- Loading and preprocessing the training images

```
image_list, label_list = [], []
# ...
```

MODEL TRAINING AND EVALUATION

- Transforming the image data and labels :

```
np_image_list = np.array(image_list,
dtype=np.float16) / 225.0
```

- Splitting the data into training and validation sets:

```
x_train, x_test, y_train, y_test =
train_test_split(np_image_list, image_labels,
test_size=0.2, random_state = 42)
```

- Defining the model architecture:

```
model = Sequential()
#..............

model.summary()
```

- Training the model and obtaining accuracy/loss values

```
history = model.fit_generator(augment.flow(x_tr……
```

RESULTS AND INTERPRETATION

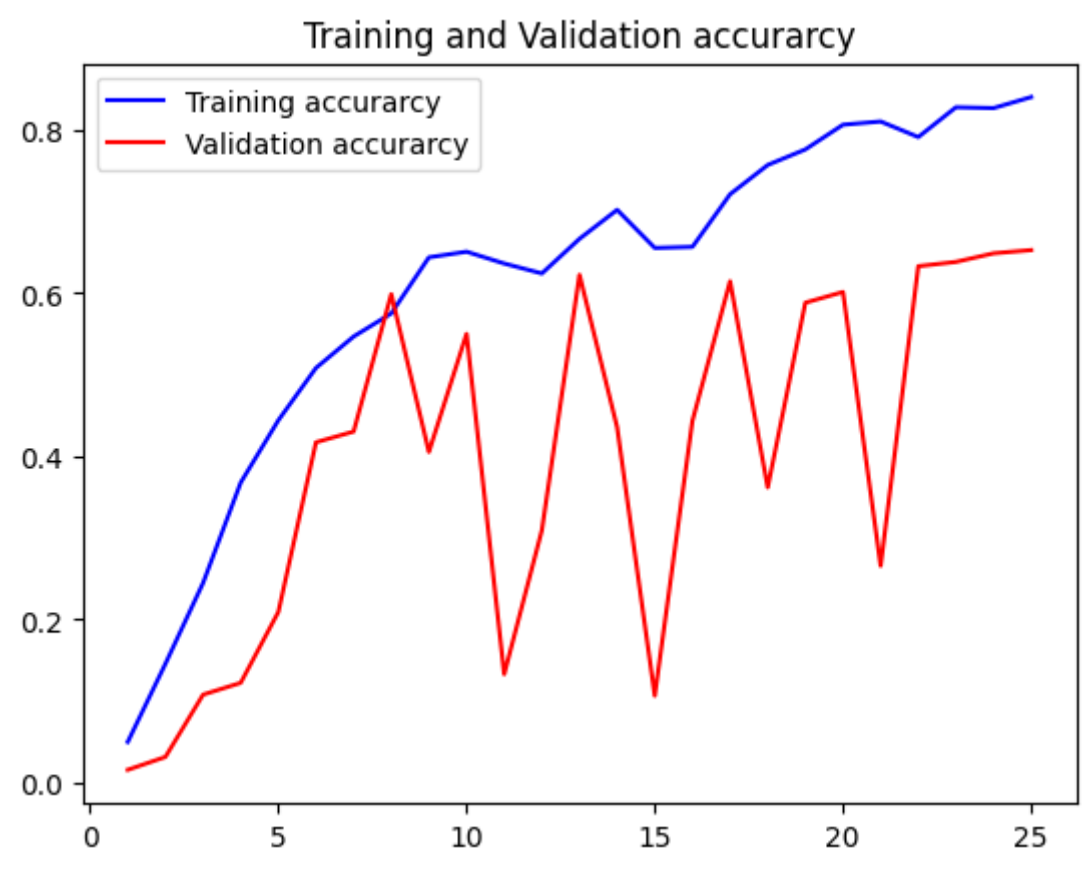
- Visualizing the training and validation accuracy

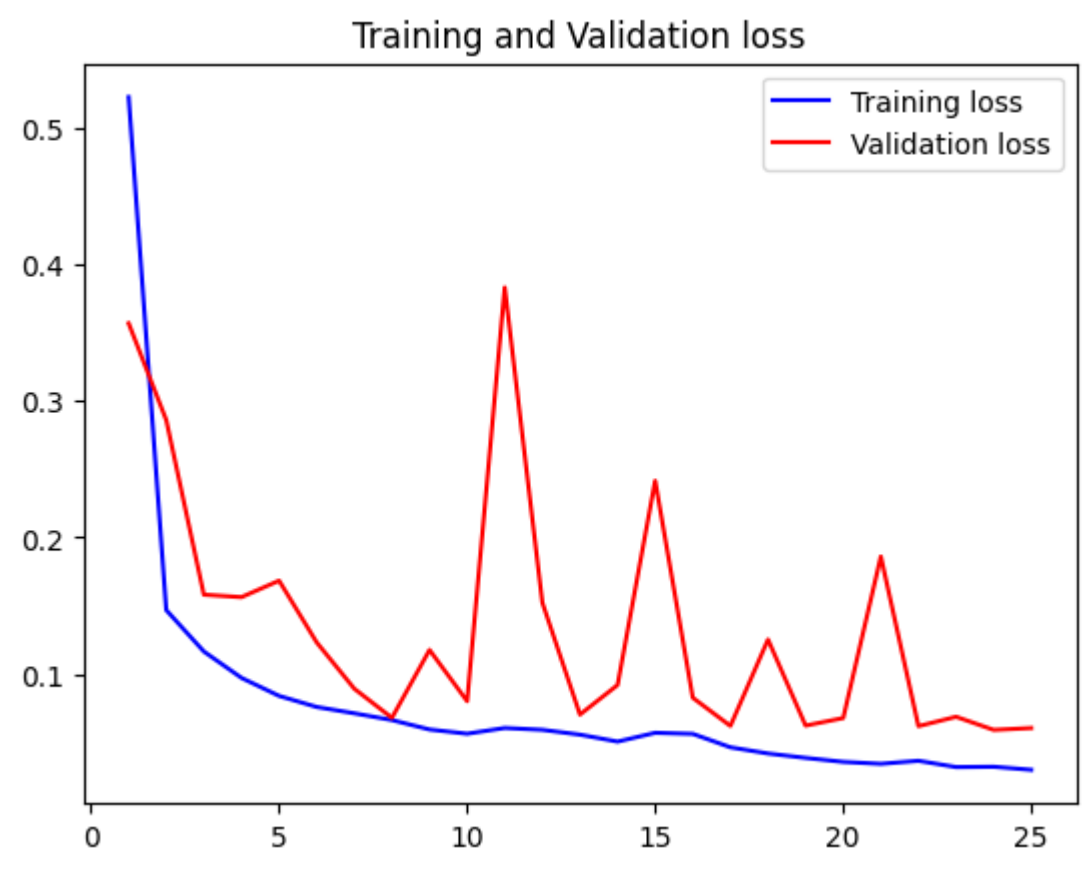
```
plt.plot(epochs, acc, 'b',
label='Training accurarcy')
plt.plot(epochs, val_acc, 'r',
label='Validation accurarcy')
plt.title('Training and Validation
accurarcy')
plt.legend()
plt.figure()
```

- Visualizing the training and validation loss

```
plt.plot(epochs, loss, 'b',
label='Training loss')
plt.plot(epochs, val_loss, 'r',
label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```

VISUALIZING ACCURACY AND LOSS





SAVING MODEL

```
# Dump pickle file of the model
print("[INFO] Saving model...")
pickle.dump(model,open('plant_disease_classi
fication_model.pkl', ‘wb'))

# Dump pickle file of the labels
print("[INFO] Saving label transform...")
filename =
'plant_disease_label_transform.pkl'
image_labels = pickle.load(open(filename,
‘rb'))
```

PREDICTING PLANT DISEASES

```
def predict_disease(image_path):
    image_array =
convert_image_to_array(image_path)
    np_image = np.array(image_array,
dtype=np.float16) / 225.0
    np_image = np.expand_dims(np_image, 0)
    plt.imshow(plt.imread(image_path))
    result = model.predict(np_image)
    predicted_class_index =
np.argmax(result)
    predicted_class =
image_labels.classes_[predicted_class_index]
    print(predicted_class)
```

EXAMPLE PREDICTIONS

```
predict_disease('image_path_1.jpg')
predict_disease('image_path_2.jpg')
predict_disease('image_path_3.jpg')
```

FUTURE SCOPE AND IMPROVEMENTS.

- Expanding the Dataset
- Object Detection
- Transfer Learning
- Ensemble Learning
- Online Learning and Continuous Improvement

THANK YOU