LEAF DISEASE DETECTION

Using Deep Learning for Accurate Plant Disease Diagnosis

Introduction

Overview:

- In modern agriculture, early detection of plant diseases is crucial for ensuring crop health and maximizing yield. However, traditional methods of disease detection are time-consuming and require expert knowledge.

Project Objective:

- This project aims to develop an automated system for detecting plant diseases using deep learning techniques. By analyzing images of leaves, our system can accurately classify various diseases, enabling timely and effective interventions.

Introduction

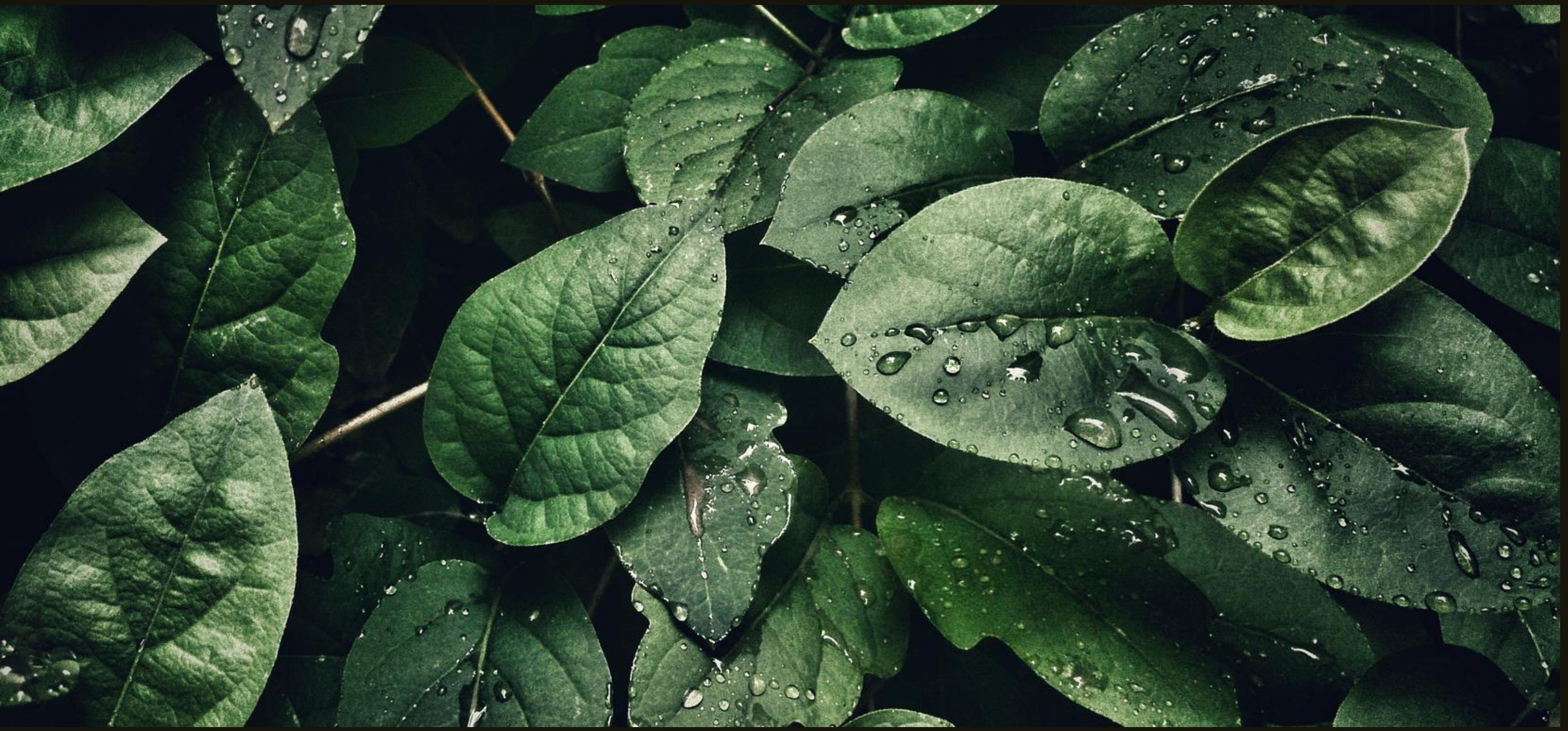
Why Deep Learning?

- Deep learning models, particularly Convolutional Neural Networks (CNNs), are powerful tools for image recognition tasks. They can automatically learn and extract features from images, making them ideal for disease detection in plants.

Key Benefits:

- Efficiency: Provides rapid diagnosis without the need for expert intervention.
- Accuracy: High accuracy in disease classification ensures reliable results.
- Scalability: Can be deployed across different crops and environments with minimal adjustment.

Project Highlights



Data Collection:

Utilized a comprehensive dataset of leaf images from various plant species and disease types.

Model Architecture:

Built a Convolutional Neural Network (CNN) model with multiple layers to extract and classify features.

Performance:

Achieved a training accuracy of 97.82% and a validation accuracy of 94.59%.

Impact:

Provides a scalable solution for improving crop management and disease prevention.

Data Preprocessing

```
training_set = tf.keras.utils.image_dataset_from_directory
('dataset/train' , labels= "inferred" , label_mode = "categorical" ,
   image_size = (128, 128) , batch_size = 32 , shuffle = True)

validation_set = tf.keras.utils.image_dataset_from_directory
('dataset/valid' , labels = "inferred" , label_mode = "categorical" ,
   image_size = (128, 128) , batch_size= 32 , shuffle = True)
```

Model Building

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=3,
activation= 'relu’ , input_shape=[128, 128, 3]))

model.add(MaxPool2D(pool_size=2, strides=2))

# Additional convolutional layers
model.add(Dropout(0.25))
model.add(Flatten())

model.add(Dense(units=1500, activation= 'relu’ ))

model.add(Dense(units=38, activation= 'softmax’ ))

model.add(Dropout(0.4))
```

Model Training and Evaluation

```
training_history =
model.fit(training_set , validation_data =
validation_set , epochs = 10)

train_loss , train_acc =
model.evaluate(training_set)

val_loss , val_acc =
model.evaluate(validation_set)
```

Model Testing

```
image_path = 'dataset/test/AppleCedarRust1.JPG’

img = cv2.imread(image_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)

image =
tf.keras.preprocessing.image.load_img(image_path,
target_size=(128,128))

input_arr =
tf.keras.preprocessing.image.img_to_array(image)

input_arr = np.array([input_arr])
predictions = model.predict(input_arr)
result_index = np.argmax(predictions)
```

Model Performance Metrics

```
from sklearn.metrics import classification_report,
confusion_matrix

print(classification_report(y_true,
predicted_categories, target_names=class_name))

cm = confusion_matrix(y_true, predicted_categories)
sns.heatmap(cm, annot=True, annot_kws={"size":10})
```

THANK YOU!

(+20)1207605310

abdelrahmanemad2530@gmail.com