Plant Leaf Disease Detection Using CNN

by

Mohammad Sabbir Ahmed,	ID: 17183103004
Syeda Nowshin Ibnat,	ID: 17183103020
Rakibul Ahasan,	ID: 17183103022
Nusrat Jahan Anka,	ID: 17183103008
Sk. Abu Hanif,	ID: 17183103043

Submitted in partial fulfilment of the requirements of the degree of
Bachelor of Science in
Computer Science and Engineering

Department of Computer Science and Engineering
Bangladesh University of Business and Technology

July 27, 2022

APPROVAL

We do hereby acknowledge that the project entitled "Plant Leaf Disease Detection Using CNN" was carried out by Mohammad Sabbir Ahmed, Syeda Nowshin Ibnat, Rakibul Ahasan, Nusrat Jahan Anka and Sk. Abu Hanif ID No: 17183103004, 17183103020, 17183103022, 17183103008, and 17183103043 Department of CSE, Bangladesh University of Business and Technology (BUBT) under the supervision of Badhan Chandra Das, Lecturer, Department of Computer Science and Engineering (CSE), BUBT. We declare that no part of this work has been submitted elsewhere for the requirements of any degree, award or diploma, or any other purposes except for publications.

Badhan Chandra Das

Lecturer

Department of Computer Science and Engineering (CSE)

Bangladesh University of Business and Technology (BUBT)

Mirpur-2, Dhaka-1216, Bangladesh

ACKNOWLEDGEMENTS

We would like to thank the following people for their help in the production of this project Badhan Chandra Das, project supervisor for all of his assistance with the project. And we would like to thank the CSE department faculty members for their support and guidance which helped us to nurture our knowledge and skill.

ABSTRACT

Agricultural productivity is a key component of the Bangladesh economy. Therefore the contribution of food crops and cash crops is highly important for both the environment and human beings. Every year crops succumb to several diseases. Due to inadequate diagnosis of such diseases and not knowing symptoms of the disease and its treatment many plants die. This study provides insights into an overview of plant disease detection using different algorithms. A CNN-based method for plant disease detection has been proposed here. Simulation study and analysis are done on sample images in terms of time complexity and the area of the infected region. It is done by image processing technique. A total of 38 cases have been fed to the model, Test accuracy is obtained as 77.64%.

Table of Contents

Introduction

A convolutional neural network (CNN) is a type of artificial neural network used primarily for image recognition and processing, due to its ability to recognize patterns in images. CNN requires very little pre-process data as compared to other deep learning algorithms. As we know the primary occupation in Bangladesh is agriculture. Here in Bangladesh, farmers cultivate a great diversity of crops. Various factors such as climatic conditions, soil conditions, various disease, etc affect the production of the crops. The existing method for plants disease detection is simply an observation which requires more man labor, properly equipped laboratories, expensive devices ,etc. The problem of efficient disease protection is closely associated with the problems of sustainable agriculture Inexperienced pesticide usage can cause the event of long-term resistance of the pathogens, severely reducing the power to fight back. Timely and accurate diagnosis of plant diseases is one among the pillars of precision agriculture. It is crucial to stop unnecessary waste of monetary and other resources, thus achieving healthier production during this changing environment, appropriate and timely disease identification including early prevention has never been more important. We are proposing this model as an alternative to the traditionally time-consuming process, different research works plan to find feasible approaches towards protecting plants.

Contribution

The main contribution is to identify the plant diseases using image processing. It completes each of the process sequentially and hence achieving each of the output. Thus the others are:

- Design a system that can detect crop disease accurately.
- Create a database of insecticides for respective disease.

Dataset Details

Form total collected data we have elected around 60343 valid data and the dataset separated into several folders. The dataset which is selected as training and testing and its ratio is 80 percent and 20 percent respectively. The testing section have been selected 103 images for the disease of Foliar fungal, 239 images for Gray_leaf_spot, Rust disease select 239 images,

233 images for Common_rust_corn, maize healthy.images 233, sort hole disease select 459 images, peach healthy image number is 72, Alternaria_blight disease select 80 images, Anthracnose select 80 images, for downy mildew 94 images 80 healthy images, powdery mildew select 80 images, Black_rot uses 233 images. During training portion of the work Foliar_Fungal uses 410 images, select 953 images as Gray leaf spot disease, Rust disease collect 929 images, 929 maize healthy images, Short_hole disease elected 1838 images, peach fruits use 288 healthy images. The dataset which is collected before starting the research work is reshaped duo to match the picture size with each other and set the pixel size as 256×256.

The example of our selected dataset which occur before our work starting is given below as Fig 1.

Fig1: Accumulated Dataset

```
{0: 'Apple___Apple_scab',
 1: 'Apple___Black_rot',
 2: 'Apple___Cedar_apple_rust',
 3: 'Apple___healthy',
 4: 'Blueberry___healthy',
 5: 'Cherry___Powdery_mildew',
 6: 'Cherry___healthy',
 7: 'Corn___Cercospora_leaf_spot Gray_leaf_spot',
 8: 'Corn___Common_rust',
 9: 'Corn___Northern_Leaf_Blight',
 10: 'Corn___healthy',
 11: 'Grape___Black_rot',
 12: 'Grape___Esca_(Black_Measles)',
 13: 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
 14: 'Grape___healthy',
 15: 'Orange___Haunglongbing_(Citrus_greening)',
 16: 'Peach___Bacterial_spot',
 17: 'Peach___healthy',
 18: 'Pepper,_bell___Bacterial_spot',
 19: 'Pepper,_bell___healthy',
 20: 'Potato___Early_blight',
 21: 'Potato___Late_blight',
 22: 'Potato___healthy',
 23: 'Raspberry___healthy',
 24: 'Soybean___healthy',
 25: 'Squash___Powdery_mildew',
 26: 'Strawberry___Leaf_scorch',
 27: 'Strawberry___healthy',
 28: 'Tomato___Bacterial_spot',
 29: 'Tomato___Early_blight',
 30: 'Tomato___Late_blight',
 31: 'Tomato___Leaf_Mold',
 32: 'Tomato___Septoria_leaf_spot',
 33: 'Tomato___Spider_mites Two-spotted_spider_mite',
 34: 'Tomato___Target_Spot',
 35: 'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
 36: 'Tomato___Tomato_mosaic_virus',
 37: 'Tomato___healthy'}
```

Fig 2: Classes of plant leaf diseases

Proposed System

In this project, a deep learning-based model is proposed to solve the problem. The deep neural layer is trained using a public dataset containing images of healthy and diseased crop leaves. The model serves its objective by classifying images of leaves into diseased categories based on the pattern of the defect. The leaves have texture and visual similarities which are attributes for the identification of disease type. Hence, computer vision employed with deep learning provides the way to solve this problem.

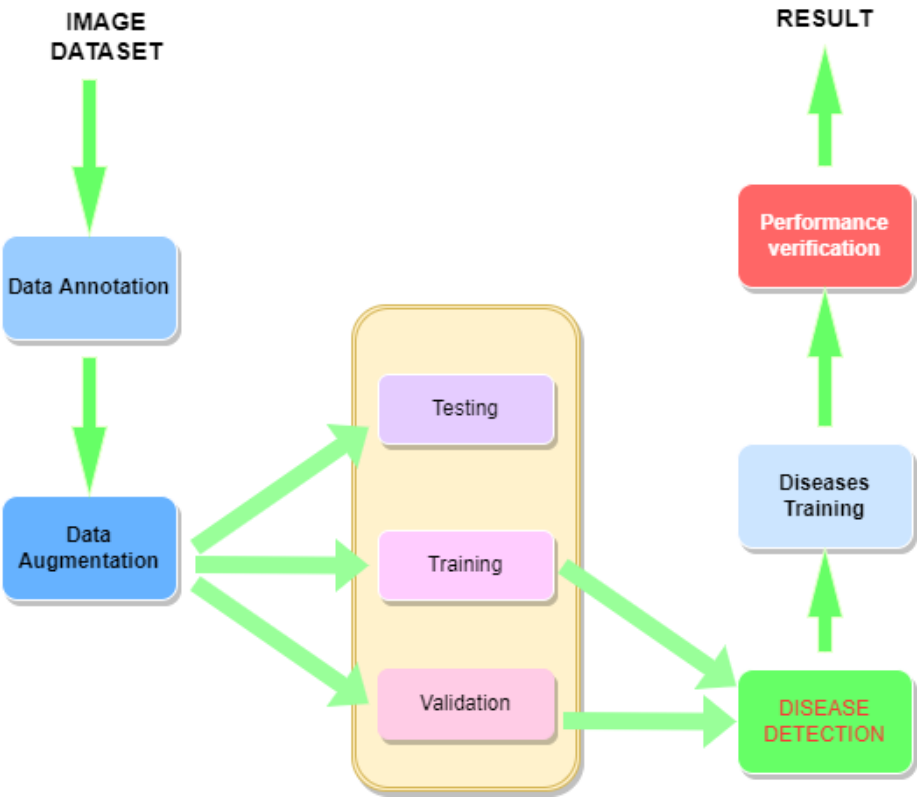

Fig 3 : Diagram of the proposed system

Experimental Setup

We did this project on “Google Colab”. Some snaps are given below:

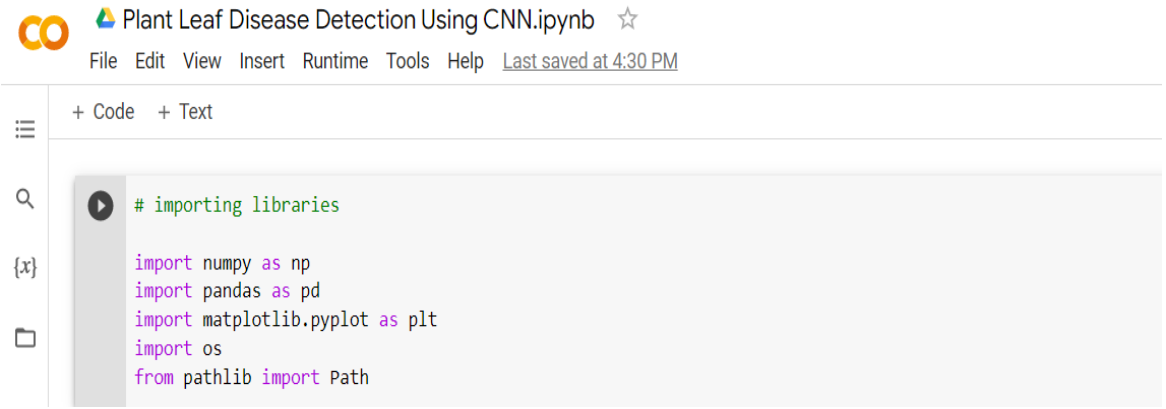

Fig 4: Importing all the needed libraries

```
Dim = 256 # dimension of the image
batch_size = 32 #batch size specified to be trained
Num_class = 39 #number of classes of the dataset
default_image_size = tuple((256, 256))
```

```
# 0.2 basicallmy means,20 out 100 images will be used for validation and 80 of them will be used for training.
datagen = ImageDataGenerator(zoom_range = 0.5,
                             shear_range=0.3,
                             horizontal_flip=True,
                             validation_split = 0.2,
                             preprocessing_function=preprocess_input)
```

```
# train_gen and val_gen is a set of images which will be called in later aspect of our code.
train_gen = datagen.flow_from_directory(directory="/content/Plant_leaf_diseases_dataset",
                                        target_size = (Dim, Dim),
                                        batch_size = 32)

val_gen = datagen.flow_from_directory(directory="/content/Plant_leaf_diseases_dataset",
                                      target_size = (Dim, Dim),
                                      batch_size = 32)
```

```
Found 60343 images belonging to 38 classes.
Found 60343 images belonging to 38 classes.
```

```
def plotImage(img_arr,label):
   for im ,l in zip(img_arr,label):
     plt.figure(figsize=(5,5))
     plt.imshow(im/255)
     plt.show()
```

```
plotImage(t_img[:3],label[:3])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

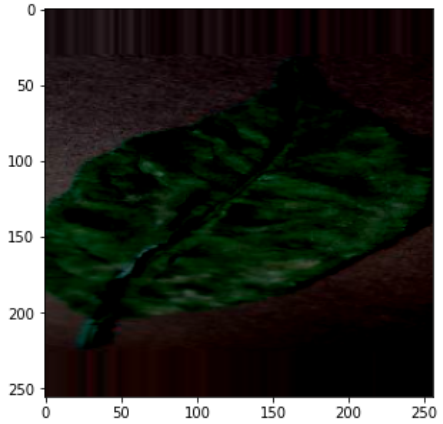

Fig 5 : Data Processing

building our model

```
[ ] from keras.layers import Dense ,Flatten
    from keras.models import Model
    from keras.applications.vgg19 import VGG19
    import keras
```

```
[ ] base_model = VGG19(input_shape=(Dim,Dim,3),include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80142336/80134624 [==============================] - 0s 0us/step
80150528/80134624 [==============================] - 0s 0us/step
```

```
[ ] for layer in base_model.layers:
      layer.trainable = False
```

```
[ ] base_model.summary()
```

```
Model: "vgg19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0

 block1_conv1 (Conv2D)       (None, 256, 256, 64)      1792

 block1_conv2 (Conv2D)       (None, 256, 256, 64)      36928

 block1_pool (MaxPooling2D)  (None, 128, 128, 64)      0

 block2_conv1 (Conv2D)       (None, 128, 128, 128)     73856

 block2_conv2 (Conv2D)       (None, 128, 128, 128)     147584

 block2_pool (MaxPooling2D)  (None, 64, 64, 128)       0

 block3_conv1 (Conv2D)       (None, 64, 64, 256)       295168
```

Fig 6: Model being trained

Result

As it is known that convolutional networks are ready to learn features when trained on larger datasets, results achieved when trained with only original images will not be explored. After fine-tuning the parameters of the network, an overall accuracy of 77.64% was achieved.

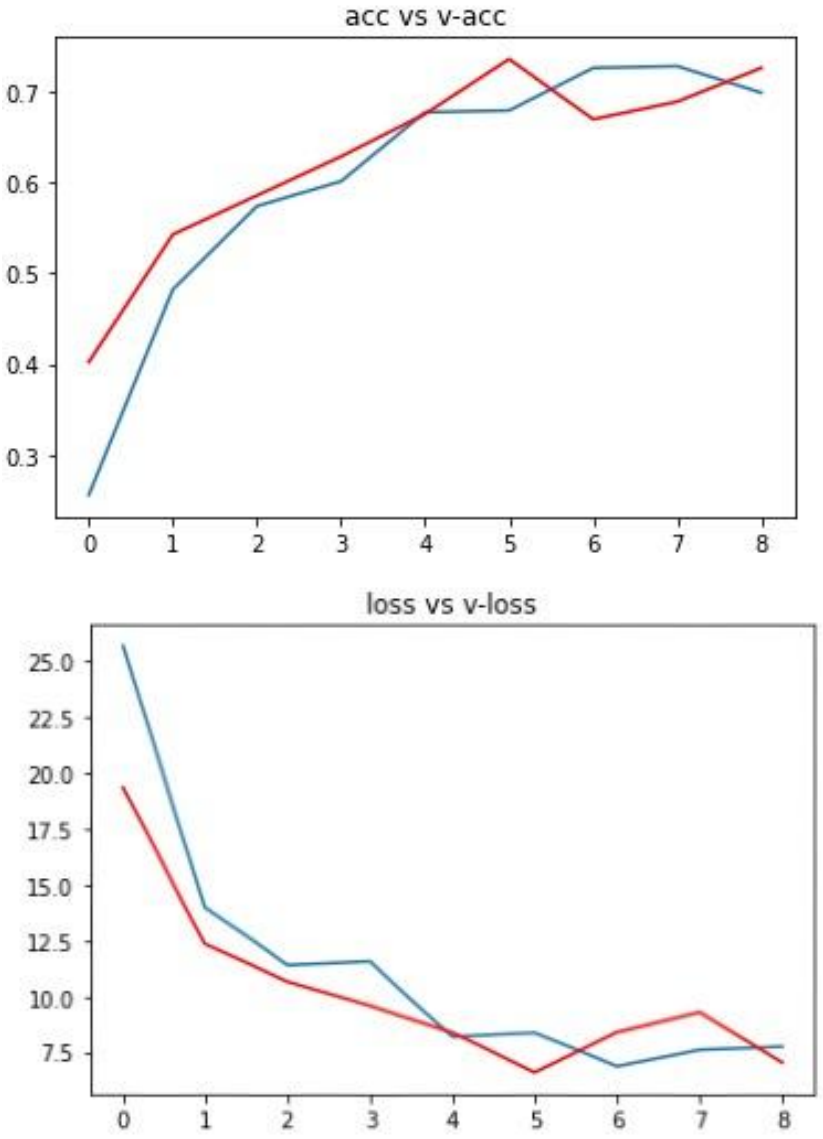

Fig 7: Training and validation accuracy and loss graph.

```
def prediction(path):

  img = load_img(path, target_size=(256,256))
  i = img_to_array(img)
  im = preprocess_input(i)
  # print(im)
  # print(im.shape)

  img = np.expand_dims(im, axis=0)
  # print(img.shape)

  pred = model.predict(img)
  pred = np.argmax(model.predict(img))
  print(pred)

  print(f"The image belongs to { ref[pred] }\n")
```

```
for images in os.listdir(test_data_root):
  showImage('/content/test_data/'+images)
  prediction('/content/test_data/'+images)
```

```
32
The image belongs to Tomato___Septoria_leaf_spot
```

Fig 8: The output is tomato_septoria_leaf_spot.

Conclusion

The proposed algorithm is implemented successfully to train the system. The accuracy percentage on the test set is 77.64% with no overfitting There is still room for improvement as the remaining 22.36% is covered. This present work can contribute to agricultural domain and can be used to help people for tracking their house plants and also enables the farmers to keep a track of the harvest. This work can be expanded into further to develop an app through which one would also know the remedy to a plant disease.

Reference

[1] **https://www.geeksforgeeks.org/python-data-augmentation/**

[2] **https://keras.io/api/callbacks/early_stopping/**

[3] **https://www.kaggle.com/**