Project Report
on

Plant Disease Detection using Transfer Learning (MobileNet v2)

(A dissertation submitted in partial fulfillment of the requirements of Bachelor of Technology in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology, West Bengal)

Submitted by

Shreya Kundu - IT - 11200219007

Under the guidance of

Prof./Dr. Kalyan Mahata
Professor/Asst. Prof./Lecturer,
Dept. of Computer Science and Engineering

Government College of Engineering and Leather Technology
(Affiliated to MAKAUT, West Bengal)
Kolkata - 700106, WB
2022

Certificate of Approval

This is to certify that the project report on “Plant Disease Detection using Transfer Learning(MobileNet v2)” is a record of bonafide work, carried out by Shreya kundu, Souhitya Biswas, Rahul Acharyya, Sreetama Goswami and Rohit Dey under the guidance and supervision of Prof./Dr. Kalyan Mahata.

In my opinion, the report in its present form is in conformity with the specified by the Government College of Engineering and Leather Technology and as per regulations of the Maulana Abul Kalam Azad University of Technology, West Bengal. To the best of my knowledge, the results presented here are original in nature and worthy of incorporation in the project report for the B.Tech. Program in Computer Science and Engineering.

Signature of
Supervisor/ Guide

Signature of
Head, Dept. of IT

ACKNOWLEDGEMENT

With great pleasure, I would like to express my profound gratitude and indebtedness to Prof./Dr. Kalyan Mahata, Department Of Information Technology, Government College of Engineering and Leather Technology, W.B. for his continuous guidance, valuable advice, and constant encouragement throughout the project work. His valuable and constructive suggestions in many difficult situations are immensely acknowledged. I am short of words to express his contribution to this thesis through criticism, suggestions, and discussions.

I would like to take this opportunity to thank Prof./Dr. Kalyan Mahata, Project Coordinator, and Prof. Santanu Halder, HOD, Department of Computer Science & Engineering and Information Technology, Government College of Engineering and Leather Technology.

SIGNATURE:

Shreya Kundu- 11200219007	**Souhitya Biswas- 11200219015**
Rahul Acharyya- 11200219017	**Sreetama Goswami- 11200219018**
Rohit Dey- 11200219021	

CONTENTS

ABSTRACT

Agriculture plays a vital role in India's economy. Over 60 percent of the rural households depend on agriculture as their principal means of livelihood. However, the farmers of India have been facing a lot of real time challenges like crop diseases. They are major threat to food security, but their immediate identification remains difficult in many places due to the lack of proper facilities.

However, increasing smartphone penetration along with advancement in computer vision that is made possible via deep learning, have paved the way for smartphone-assisted disease diagnosis. Using a public dataset of about 87K RGB images of healthy and diseased crop leaves which are categorized into 38 different classes under controlled conditions, we train a deep convolutional neural network to identify 14 crop species. The trained model achieves an accuracy between 90-99% on the testing dataset.

The MobileNetV2 architecture is compatible with mobile devices using the optimized parameter. The accuracy results in the identification of diseases showed that the deep CNN model is promising and can greatly impact the efficient identification of the diseases, and may have potential in the detection of diseases in real-time agricultural systems.

1. INTRODUCTION

Agriculture plays an important role in the Indian economy. Plant disease identification is the key to preventing losses in yield as well as agricultural product quantity. On the other hand, some factors like temperature, sunlight, soil moisture, pathogens, and nutrition affect the growth of plants. Among those factors, pathogens are the major factor that leads to the destruction of the plant. To eradicate these pathogens, farmers use pesticides. Without identifying the type of pathogen, it is not advised to use any kind of pesticide.

1.1 PROBLEM STATEMENT

Plant diseases are impacting agriculture in general and monoculture more precisely. The traditional way to deal with the detection of plant diseases plants disease detection is by visual inspection of the leaves or some chemical processes by experts.

This is time-consuming and costs a lot for the farmers. This operation of detecting plant diseases is expensive since the framers must consult experts, and time-consuming as those experts need time to detect the disease and classify its type and need continuous control.

Knowing that some farmers could not have the capability to consult the experts regularly, so the risk of contagion between plants is extremely high. Also, this will have bad impacts on the environment due to the extensive use of chemicals and pesticides that are used in random quantities, also, the production process is affected.

1.2 OBJECTIVE

- ➢ To identify various diseases in plants.
- ➢ To implement a method for preventing disease and providing help for reducing the loss or damage caused by diseases.

1.3 SYSTEM REQUIREMENTS

I. Hardware Requirements

- Processor: Intel(R) Xeon(R) CPU @ 2.00GHz
- RAM: 12GB
- System type: 64-bit operating system, x64-based processor
- SSD: 80TB
- GPU: NVIDIA Tesla T4

II. Software Requirements

- Operating System: Ubuntu 18.04 LTS
- Programming Language: Python 3.7.15
- Browser: Google Chrome
- Jupyter Notebook: Google Colab

2. APPROACH

We wanted to make the training process quicker and the output weights file to be light-weight so that in the future, we can easily adapt it to mobile devices. India (and many other countries) have seen a boom in mobile phone sales and it has penetrated the tier-3 cities of our country as well. After some research, we found that our approach should be Transfer learning based and the best model would be MobileNet pre-trained on imagenet dataset because of its ability to train fast and produce light enough to be used in mobile devices and still give us high accuracy.

2.1 Transfer Learning

Transfer learning is the reuse of a pre-trained model on a new problem. It's currently very popular in deep learning because it can train deep neural networks with comparatively little data. Transfer Learning helps us train a pre-trained model with some extra added layers. Models used for plant disease detection is MobileNet.

2.2 What is MobileNet?

MobilenetV2 is the second version of the Mobilenet model. It significantly has a lower number of parameters in the deep neural network. This results in more lightweight deep neural networks. Being lightweight, it is best suited for embedded systems and mobile devices.

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. This CNN architecture was conceived to optimize the number of operations.

MobilenetV2 is a pre-trained model for image classification. Pre-trained models are deep neural networks that are trained using a large image dataset.

3. METHODOLOGY

3.1 LIBRARIES USED

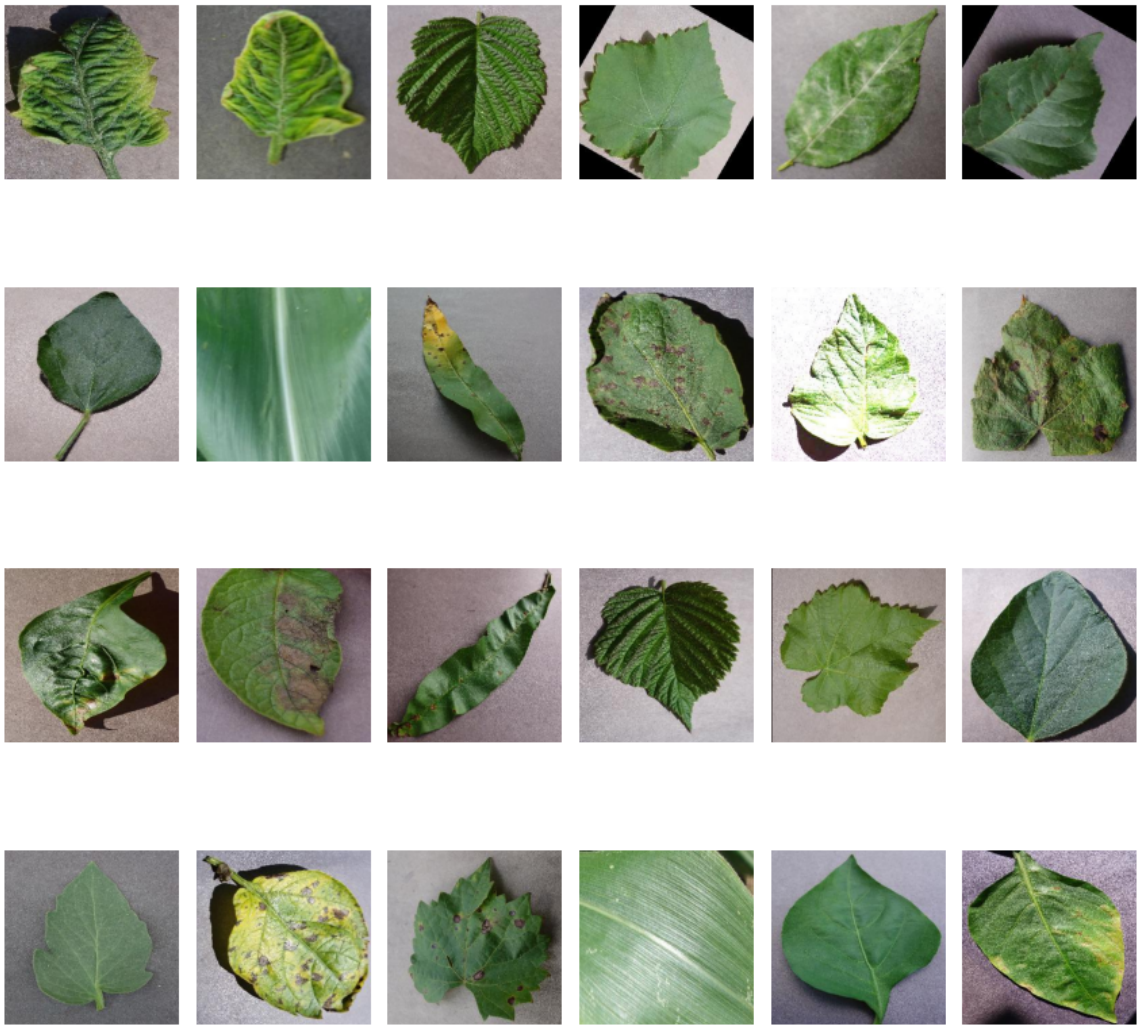
1. **TensorFlow Keras** - It is an open-source library used to develop machine learning and deep learning models. It trains deep neural networks with input, hidden, and output layers.We used TensorFlow to add custom layers to the pre-trained MobilenetV2. This will help to fine-tune the plant disease classification model and improve its performance.

2. **Matplotlib.pylab and Seaborn -** We will use the library to plot diagrams and visualization of our image dataset. It will show the model prediction results.

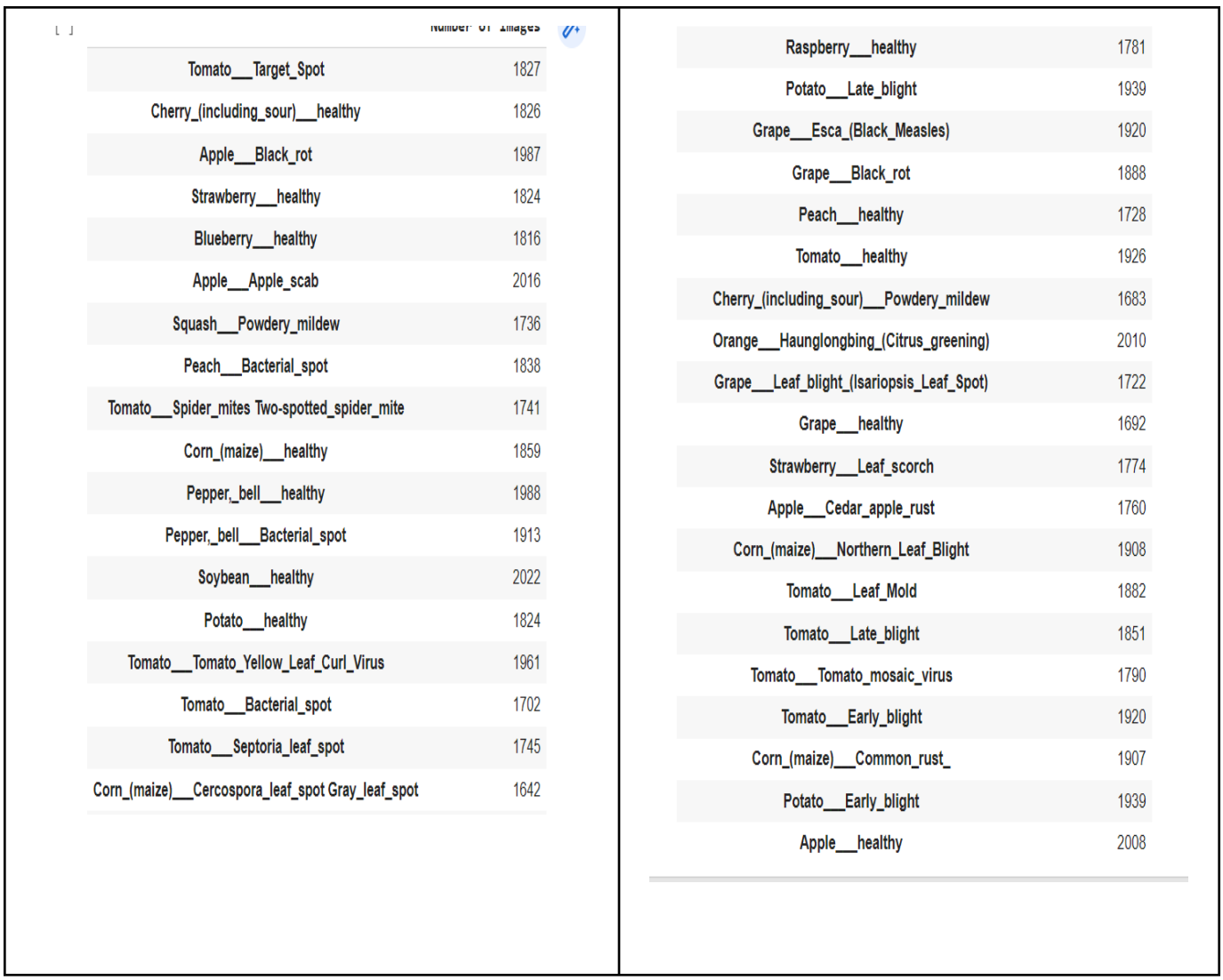
3. **Numpy and Pandas -** For handling images and dataframes.

3.2 DATASET DESCRIPTION

This dataset is recreated using offline augmentation from the original dataset. This dataset consists of about 87K RGB images of healthy and diseased crop leaves which are categorized into 38 different classes. The total dataset is divided into an 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purposes.



Sample images in dataset



Tomato___Target_Spot	1827
Cherry_(including_sour)___healthy	1826
Apple___Black_rot	1987
Strawberry___healthy	1824
Blueberry___healthy	1816
Apple___Apple_scab	2016
Squash___Powdery_mildew	1736
Peach___Bacterial_spot	1838
Tomato___Spider_mites Two-spotted_spider_mite	1741
Corn_(maize)___healthy	1859
Pepper,_bell___healthy	1988
Pepper,_bell___Bacterial_spot	1913
Soybean___healthy	2022
Potato___healthy	1824
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1961
Tomato___Bacterial_spot	1702
Tomato___Septoria_leaf_spot	1745
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1642
Raspberry___healthy	1781
Potato___Late_blight	1939
Grape___Esca_(Black_Measles)	1920
Grape___Black_rot	1888
Peach___healthy	1728
Tomato___healthy	1926
Cherry_(including_sour)___Powdery_mildew	1683
Orange___Haunglongbing_(Citrus_greening)	2010
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1722
Grape___healthy	1692
Strawberry___Leaf_scorch	1774
Apple___Cedar_apple_rust	1760
Corn_(maize)___Northern_Leaf_Blight	1908
Tomato___Leaf_Mold	1882
Tomato___Late_blight	1851
Tomato___Tomato_mosaic_virus	1790
Tomato___Early_blight	1920
Corn_(maize)___Common_rust_	1907
Potato___Early_blight	1939
Apple___healthy	2008

38 different classes and the no. of images

3.3 PREPROCESSING THE IMAGE DATA / Data pre-processing and data augmentation

In order to make the most of our few training examples, we will "augment" them via a number of random transformations, so that our model would never see twice the exact same picture.
The images are resized to 224×224 pixels, and we perform both the model optimization and predictions on these downscaled images. We used data augmentation like shearing, zooming, horizontal flipping to increase the dataset size to almost double the original dataset size.
This helps prevent overfitting and helps the model generalize better. In Keras this can be done via the keras.preprocessing.image.ImageDataGenerator class.
. We added all these to generate more variations of the image since we have less images to train.

3.4 INITIALIZING THE BASE MODEL

We initialized the pre-trained model MobileNet V2 with a matching input size as to the pre-processed image data we have which is 224×224. Mobilenet V2 model which is pre-trained on imagenet dataset. We excluded the top layers of the pre-trained model by specifying include_top=False which is ideal for feature extraction. Then we downloaded the pre-trained model and initialized it with the given parameters.

We have added the following layers to the deep neural network.

- Global Average Pooling (GAP) layer is used to minimize overfitting by reducing the total number of parameters in the model. The GAP layer is used to reduce the spatial dimensions of a two-dimensional tensor.
- The dropout layer will handle model overfitting. It ensures the model performs well using both the train and the test images.
- The dense layer is the output layer. It has 38 neurons because the dataset has 38 predefined classes.
- We have set softmax as the activation function. Softmax converts logits into probabilities. Then the the Categorical Cross-Entropy takes the output probabilities and compares it to the actual true values and makes it one-hot encoded.

3.5 COMPILING THE MODEL

We compiled the model using the compile function. It has the following parameters:

- **model_optimizer** - It ensures that the model performs well and reduces errors that model_loss generates in training. We set the model_optimizer to Adam. The results of the Adam optimizer are generally better than every other optimization algorithms, have faster computation time, and require fewer parameters for tuning. Because of all that, Adam is recommended as the default optimizer for most of the applications.

- **model_loss** - It keeps track of the errors in the model while training. We set the model_loss to Categorical Crossentropy. Cross entropy is a loss function which gives an insight into how good the process of learning is moving ahead, lower numbers are better here.

- **model_metrics** - It checks the neural network's performance and calculates the accuracy score using the train and the test set.

3.6 TRAINING THE MODEL

The train set will train the deep neural network so that it can learn and understand plant disease classification.
The validation set will adjust and fine-tune the deep neural network parameters. This will produce an improved model with accurate results.

Finally, we trained the model and the deep neural network will run for 10 epochs for both training and validation.

```
Epoch 1/10
3514/3514 [==============================] - 1160s 326ms/step - loss: 0.4400 - accuracy: 0.8665 - val_loss: 5.0465 - val_accuracy: 0.3712
Epoch 2/10
3514/3514 [==============================] - 1104s 314ms/step - loss: 0.2262 - accuracy: 0.9295 - val_loss: 1.9179 - val_accuracy: 0.6276
Epoch 3/10
3514/3514 [==============================] - 1063s 302ms/step - loss: 0.1645 - accuracy: 0.9480 - val_loss: 0.5371 - val_accuracy: 0.8683
Epoch 4/10
3514/3514 [==============================] - 1103s 314ms/step - loss: 0.1339 - accuracy: 0.9572 - val_loss: 0.6149 - val_accuracy: 0.8677
Epoch 5/10
3514/3514 [==============================] - 1114s 317ms/step - loss: 0.1138 - accuracy: 0.9635 - val_loss: 0.5243 - val_accuracy: 0.8736
Epoch 6/10
3514/3514 [==============================] - 1133s 322ms/step - loss: 0.1007 - accuracy: 0.9679 - val_loss: 0.4099 - val_accuracy: 0.8903
Epoch 7/10
3514/3514 [==============================] - 1099s 313ms/step - loss: 0.0865 - accuracy: 0.9724 - val_loss: 1.1771 - val_accuracy: 0.9203
Epoch 8/10
3514/3514 [==============================] - 1057s 301ms/step - loss: 0.0764 - accuracy: 0.9760 - val_loss: 0.3886 - val_accuracy: 0.9054
Epoch 9/10
3514/3514 [==============================] - 1101s 313ms/step - loss: 0.0711 - accuracy: 0.9777 - val_loss: 0.2286 - val_accuracy: 0.9412
Epoch 10/10
3514/3514 [==============================] - 1110s 316ms/step - loss: 0.0640 - accuracy: 0.9797 - val_loss: 0.2444 - val_accuracy: 0.9349
```

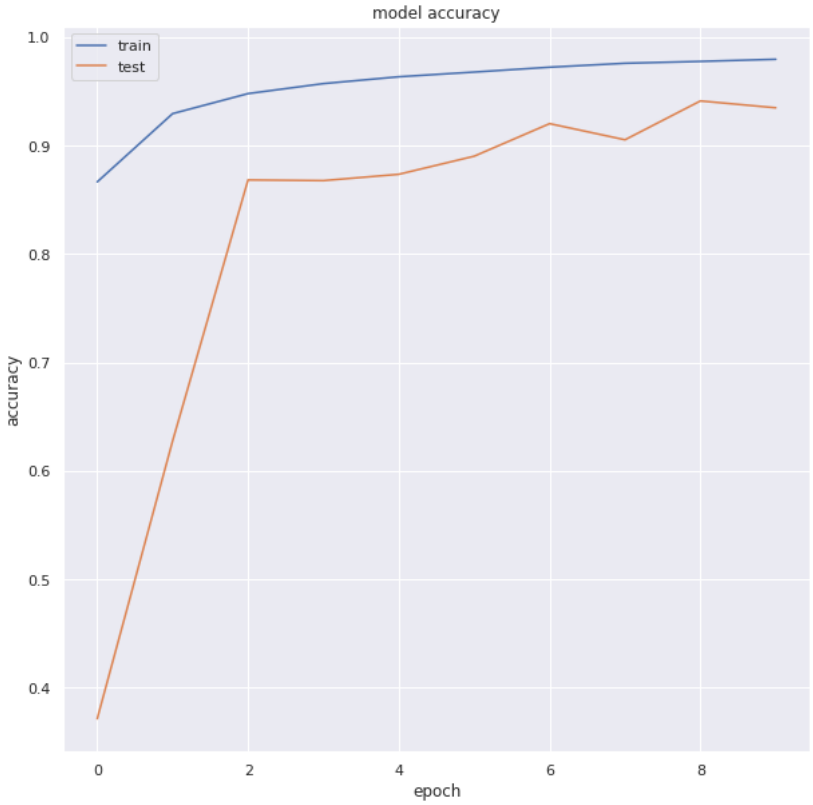
From the training process above, the first accuracy score is 0.8665 (86.65%). The last accuracy score after the 10 epochs is 0.9797 (97.97%). This shows the accuracy score increases with time.

The validation accuracy score also increases from 0.3712 (37.12%) to 0.9349 (93.49%). Moreover, the model loss reduces from 0.4400 (44.00%) to 0.0640 (6.40%). We can see the performance of the model increased with time.

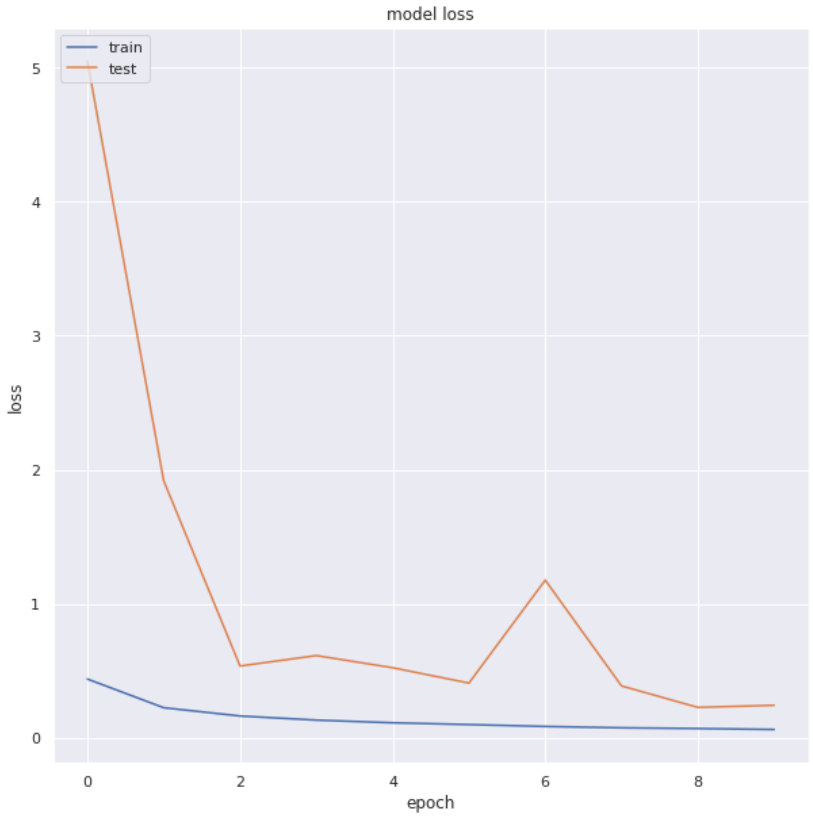
We have saved the weights for the trained model in h5 format along with the accuracy and loss in a pickle file which can be reloaded and re-used whenever required.

3.7 VISUALIZING THE TRAINING AND VALIDATION PERFORMANCE

Here, we plotted the model loss vs epochs and accuracy vs epochs graph. and confusion



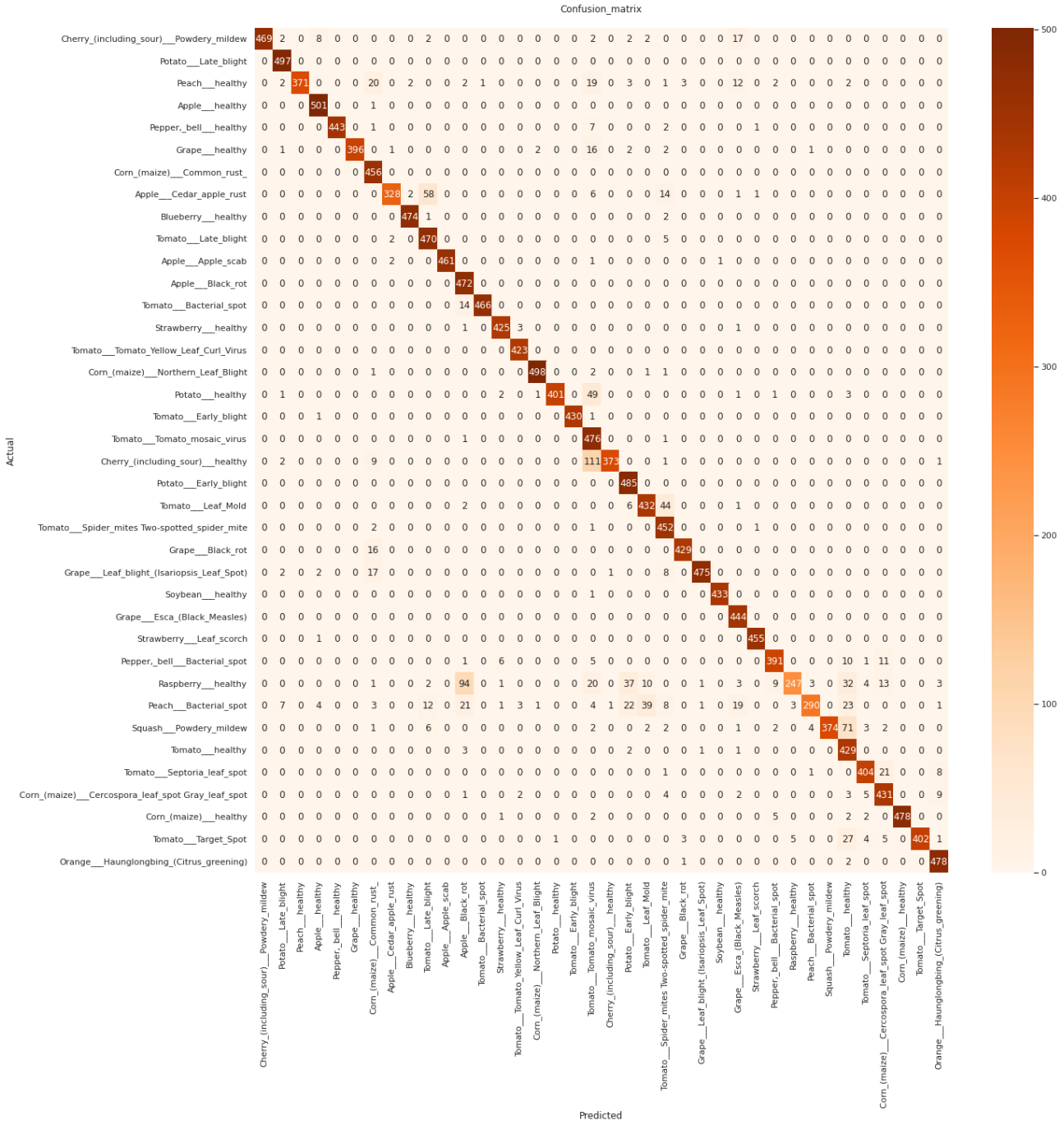
Accuracy vs Epochs graph representation



Loss vs Epochs graph representation

Next we plotted the confusion matrix for all the pairs having and not having the disease. A confusion Matrix is used as an evaluation metric when analyzing misclassification between classes. Each row of the matrix represents the class being predicted while each column represents the class which are original.

The diagonals show the classes which have been classified correctly. This tells us both which classes are being misclassified and also what they are being misclassified as.



The class-wise confusion matrix

3.8 TEST SCORE

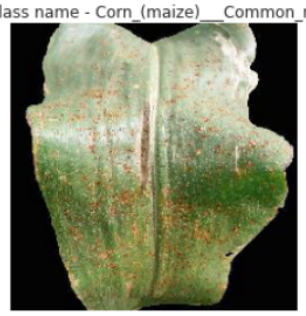
Now we test the accuracy of the model by providing new images or test data.

```
1/1 [==============================] - 0s 21ms/step
Actual class name - PotatoEarlyBlight2
Info URL - https://www.plantwise.org/knowledgebank/searchresults?q=Early%20blight
```

Predicted class name - Potato___Early_blight 100.%

```
1/1 [==============================] - 0s 20ms/step
Actual class name - CornCommonRust2
Info URL - https://www.plantwise.org/knowledgebank/searchresults?q=Common%20rust%20
```

Predicted class name - Corn_(maize)___Common_rust_ 100.%

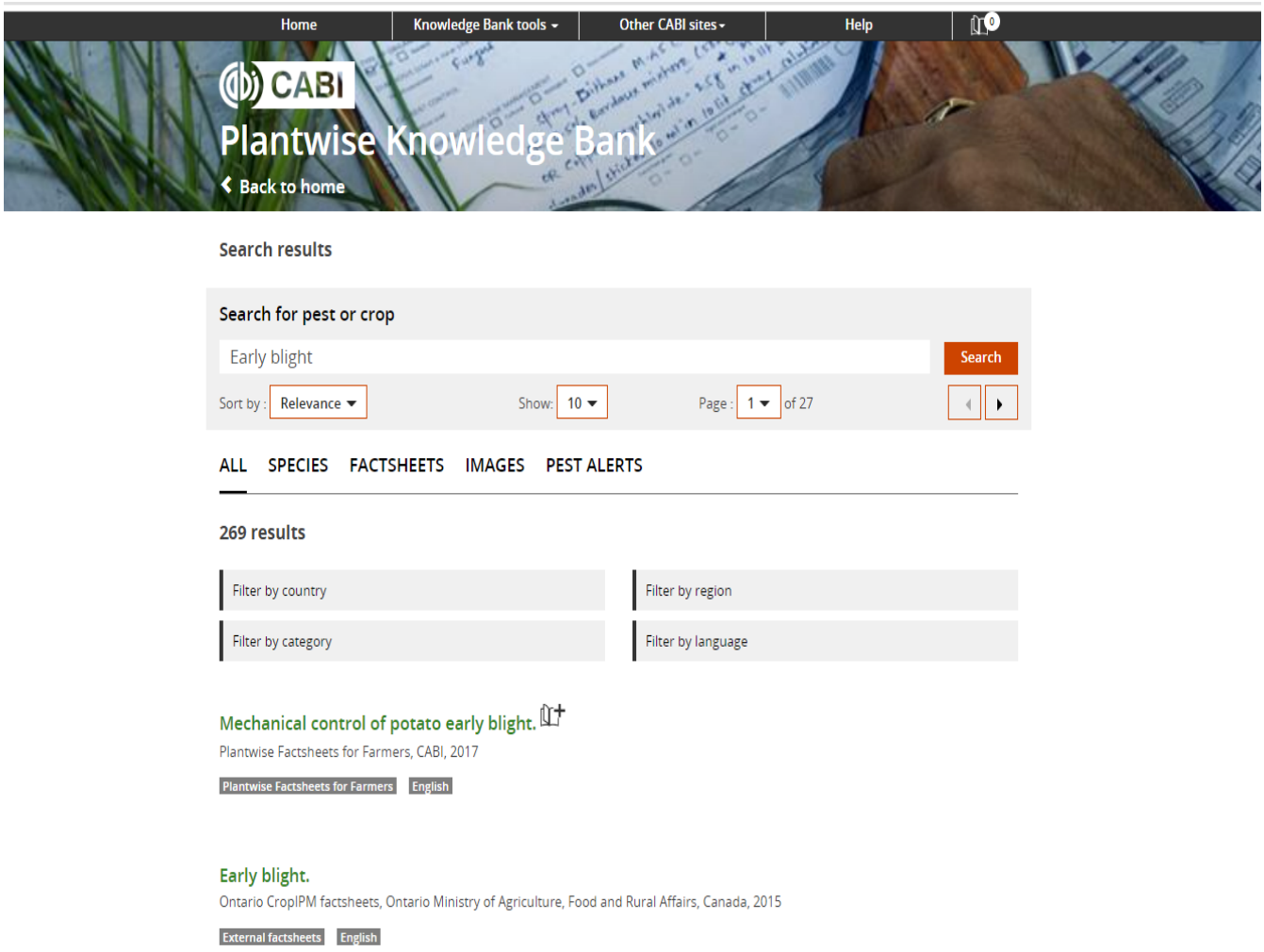


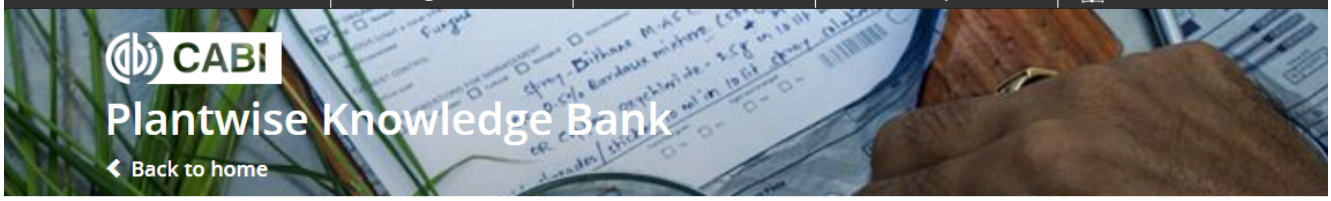
```
1/1 [==============================] - 0s 31ms/step
Actual class name - TomatoYellowCurlVirus3
Info URL - https://www.plantwise.org/knowledgebank/searchresults?q=Tomato%20Yellow%20Leaf%20Curl%20Virus
```

Predicted class name - Tomato___Tomato_Yellow_Leaf_Curl_Virus 99.9%

From this output, the actual label and the predicted label for both predictions are the same. This shows the model has made accurate predictions.

Here based on the disease detected, it will also output a URL link where farmers can go and gain knowledge about the disease, and its preventive measures like pesticides or supplements to use as per the types of disease and can also contact the nearest agriculture office.





Search results

Search for pest or crop

Common rust

Search

Sort by : Relevance | Show: 10 | Page : 1 of 30

ALL SPECIES FACTSHEETS IMAGES PEST ALERTS

291 results

Filter by country | Filter by region

Filter by category | Filter by language

Common rust.

Ontario CropIPM factsheets, Ontario Ministry of Agriculture, Food and Rural Affairs, Canada, 2015

External factsheets English

Rouille./Common rust.

Ontario CropIPM factsheets, Ontario Ministry of Agriculture, Food and Rural Affairs, Canada, 2015

External factsheets French

4. CONCLUSION

Artificial Intelligence in agriculture not only helps farmers to automate their farming but also shifts to precise cultivation for higher crop yield and better quality while using fewer resources.

We detected the disease in plants using new model architecture called MobileNets based on depthwise separable convolutions. We demonstrated MobileNet's effectiveness when applied to a wide variety of crop image datasets.

The project has shown a pretty good accuracy value of between 90% and 100%. In the future, we can build a mobile app that can provide knowledge to the farmers and suggest preventive measures like pesticides or supplements to use as per the types of disease and they can also contact the nearest agriculture office.

Moreover, the application can be extended by adding features like supporting multiple regional languages, direct integration with state departments and an option to notify and alert them about the disease so that the state is well-prepared, access to point of sale of fertilizers, pesticides/insecticides, information about crops which are disease resistant and helps in fighting it.

5. REFERENCES

1. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep feature-based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [**Google Scholar**]

2. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [**Google Scholar**]

3. Krizhevsky,A.,Sutskever,I.,&Hinton,G.E.(2012).Imagenet classification with deep convolutional neural networks.In Advances in neural information processing systems (pp. 1097-1105).

4. Andrew G. Howard , Menglong Zhu, Bo Chen, Dmitry Kalenichenko, and Weijun Wang “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv:1704.04861v1 [cs.CV] 17 Apr 2017

5. Alex Krizhevsky, IlyaSutskever, and GeoffreyE. Hinton " ImageNet Classification with Deep Convolutional Neural Networks”