# **Classification of Plant Diseases Using Deep Learning Neural Networks with EffivientnetB0**

**Durgesh Pal**

*Abstract*—Plant disease reduces the crop production. To increase the plant productivity, early detection of disease is important. A variety of deep learning models are proposed for the plant disease on the laboratory images. But this model is designed for on field large dataset .

Keywords— Convolution Neural Network, Plant Disease Classsification, Transfer Learning, Machine Learning, Deep Learning

# Introduction

Plant diseases is undesirable for agricultural production and this will results as food insecurities. For agriculture to be sustainable and proper, as well as to avoid loss of money and other resources, timely and precise detection of plant diseases is essential. Plant diseases must be prevented and controlled effectively on the basis of early identification. Also, there is possibility that the plant diseases does not have visible symptoms, hence there should be advanced analysis methods for the disease detection. It has become increasingly important in recent years to identify plant diseases.

Generally, on-site identification of diseases of plant is often done by agricultural and forestry professionals, or by farmers using their own knowledge. This subjective approach is not only arbitrary, but also time taking, difficult, and ineffective. Also there is possibility that farmers could make mistakes because of little expertise. This will result in financial losses, time losses and environmental pollution.

In order to solve these problems, the application of image processing techniques for plant disease diagnosis has become a significant research field. CNN-based models have had great results in more recent research. Basically, the main objective is the automatic identification of plant diseases in the field. The majority of earlier research focused on images of diseases that were captured in a synthetic or laboratory. As a result, the backdrop of such images is typically a plain, solid colour with just one object in the foreground. The techniques are not appropriate for real-world application systems from the natural surroundings.

In this work, the dataset is collected from three different datasets from natural environment with 20 classes of plant disease. EfficientNetB0 model based on transfer learning have been used. With data on plant diseases, the models are tested and trained. On comparing the other models with the EfficientNetB0 model, it was seems to be more effective.

# LITERATURE review

# . METHODOLOGY

## Dataset

To evaluate the performance of the model, three distinct datasets on plant diseases has been considered in this work. The datasets are Turkey plant dataset, cinnamon plant stem and branch disease dataset and sunflower fruit and leaves dataset. It is clear that the images in the collection were taken on the site and the intensity is likewise rather constant. The Turkey plant dataset is made up of 4448 images that are broken down into 15 classes that are obtained from academics working in the field of plant protection at the Agricultural Faculty of Bingol and Inonu Universities in Turkey. The Cinnamon plant stem and branch disease dataset is made up of 326 images that are broken down into 2 classes ( RoughBark,StripeCanker ) that are gathered from Cinnamon plantation in Sri Lanka. The sunflower fruit and leaves dataset is made up of 332 images that are broken down into 3 classes(Gray mold , Leaf scars, Downy mildew) and obtained from mendeley platform.

This dataset contains total 20 classes with 5106 images of plant diseases which are classified into three categories: test, train and validation.

Further detailed classification is given in the table 1.1

| Class Name | Number of images | | |
| --- | --- | --- | --- |
| Train | Test | Validation |
| **Apple Aphis spp** | 129 | 17 | 16 |
| **Apple Eriosoma lanigerum** | 292 | 38 | 36 |
| **Apple Monillia laxa** | 204 | 26 | 25 |
| **Apple Venturia inaequalis** | 506 | 64 | 63 |
| **Apricot Coryneum beijerinckii** | 880 | 111 | 110 |
| **Apricot Monillia laxa** | 68 | 9 | 8 |
| **Cancer symptom** | 60 | 9 | 7 |
| **Cherry Aphis spp** | 284 | 37 | 35 |
| **Downy mildew** | 96 | 12 | 12 |
| **Drying symptom** | 111 | 15 | 13 |
| **Gray mold** | 57 | 8 | 7 |
| **Leaf scars** | 112 | 14 | 14 |
| **Peach Monillia laxa** | 251 | 32 | 31 |
| **PeachParthenolecanium corni** | 341 | 44 | 42 |
| **Pear Erwinia amylovora** | 172 | 22 | 21 |
| **Plum Aphis spp** | 56 | 7 | 7 |
| **RoughBark** | 126 | 17 | 15 |
| **StripeCanker** | 134 | 18 | 16 |
| **Walnut Eriophyes erineus** | 55 | 8 | 6 |
| **Walnut Gnomonia leptostyla** | 144 | 18 | 18 |

Table 1.1

Some of the sample images from each class is are listed below :

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

1. Apple Aphis spp
2. Apple Eriosoma lanigerum
3. Apple Monillia laxa
4. Apple Venturia inaequalis
5. Apricot Coryneum beijerinckii
6. Apricot Monillia laxa
7. Cancer symptom
8. Cherry Aphis spp
9. Downy mildew
10. Drying symptom
11. Gray mold
12. Leaf scars
13. Peach Monillia laxa
14. Peach Parthenolecanium corni
15. Pear Erwinia amylovora
16. Plum Aphis spp
17. RoughBark
18. StripeCanker
19. Walnut Eriophyes erineus
20. Walnut Gnomonia leptostyla

## Steps

* Image Acquisition
* Preprocessing(resizing)
* Split into Test,train and Validation
* Training on data
* Validation
* Testing
* Visualization of results

## Model and Methodology

To achieve the faster processing time and higher accuracy in classification problems, deep learning models are developed. EfficientNetB0 model is used in this work. The simple and highly effective EfficientNet model uses swish as activation function instead of ReLU. The architecture of this model consists of 8 different models (B0 to B7).The proposed model have three modules to identify the plant disease: preprocessing, feature extraction and feature classification.

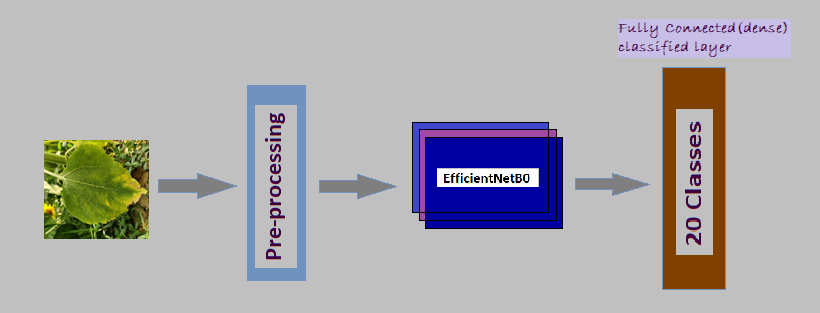


Figure : System Architecture

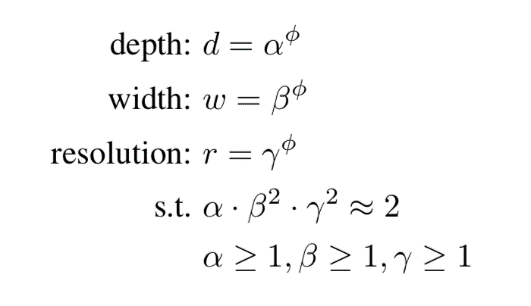
EfficientNets,as the name suggests are very much efficient computationally and also achieved state of art result on ImageNet dataset which is 84.4% top-1 accuracy. Model scaling is about scaling the existing model in terms of model depth, model width, and less popular input image resolution to improve the performance of the model. Depth wise scaling is most popular amongst all, e.g. ResNet can be scaled from Resnet18 to ResNet200. Here ResNet10 has 18 residual blocks and can be scaled for depth to have 200 residual blocks.

ResNet200 delivers better performance than ResNet18 and thus, manually scaling works pretty well. But there is one problem with traditional manual scaling method, after a certain level, scaling doesn’t improve performance. It starts to affect adversely by degrading performance.

The scaling method introduced in paper is named *compound scaling*and suggests that instead of scaling only one model attribute out of depth, width, and resolution; strategically scaling all three of them together delivers better results.

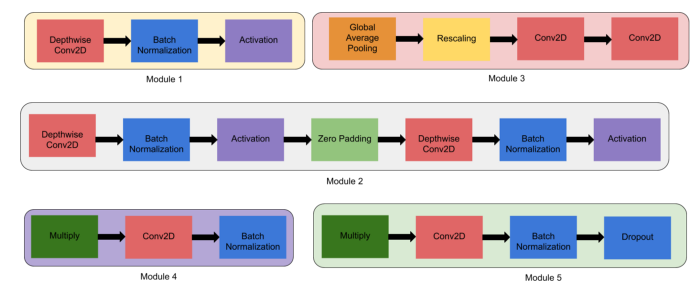
**Compound scaling**

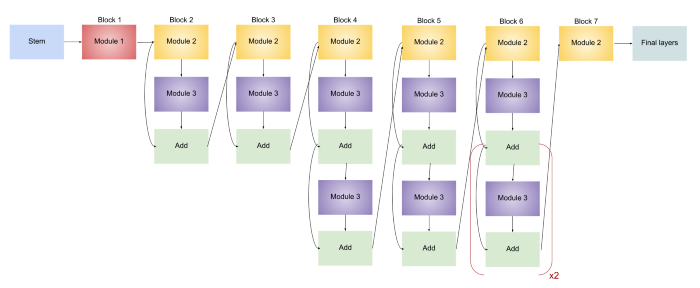
Compound scaling method uses a compound co-efficient ø to scale width, depth, and resolution together. Below is the formula for scaled attributes:



Here, alpha, beta, and gamma are scaling multiplier for depth, width and resolution respectively and be obtained using grid search. Let’s say we got alpha =1.2 after solving the above equation, then new depth = 1.2 \* old depth.

ø is a user-specific co-efficient which takes real numbers like and controls resources which is 2ø. So if we have double resources available than what a model is currently using, we can take find ø using 2ø = 2 and hence ø is 1 for such cases.

Model Architecture : 



# Experimental results

## Configuration

|  |  |
| --- | --- |
| **Parameter** | **Google Colab** |
| GPU | Nvidia K80 / T4 |
| GPU Memory | 16GB |
| GPU Memory Clock | 0.82GHz / 1.59GHz |
| Performance | 4.1 TFLOPS / 8.1 TFLOPS |

## Parameters

1. Confusion Matrix

A confusion matrix is built to evaluate how well the  model performs. The confusion matrix is a tabular representation of the effectiveness and performance of the proposed models.

For the evaluation, some terms are used such as True positive (TrPo), True negative (TrNe), False positive (FaPo), False negative (FaNe).

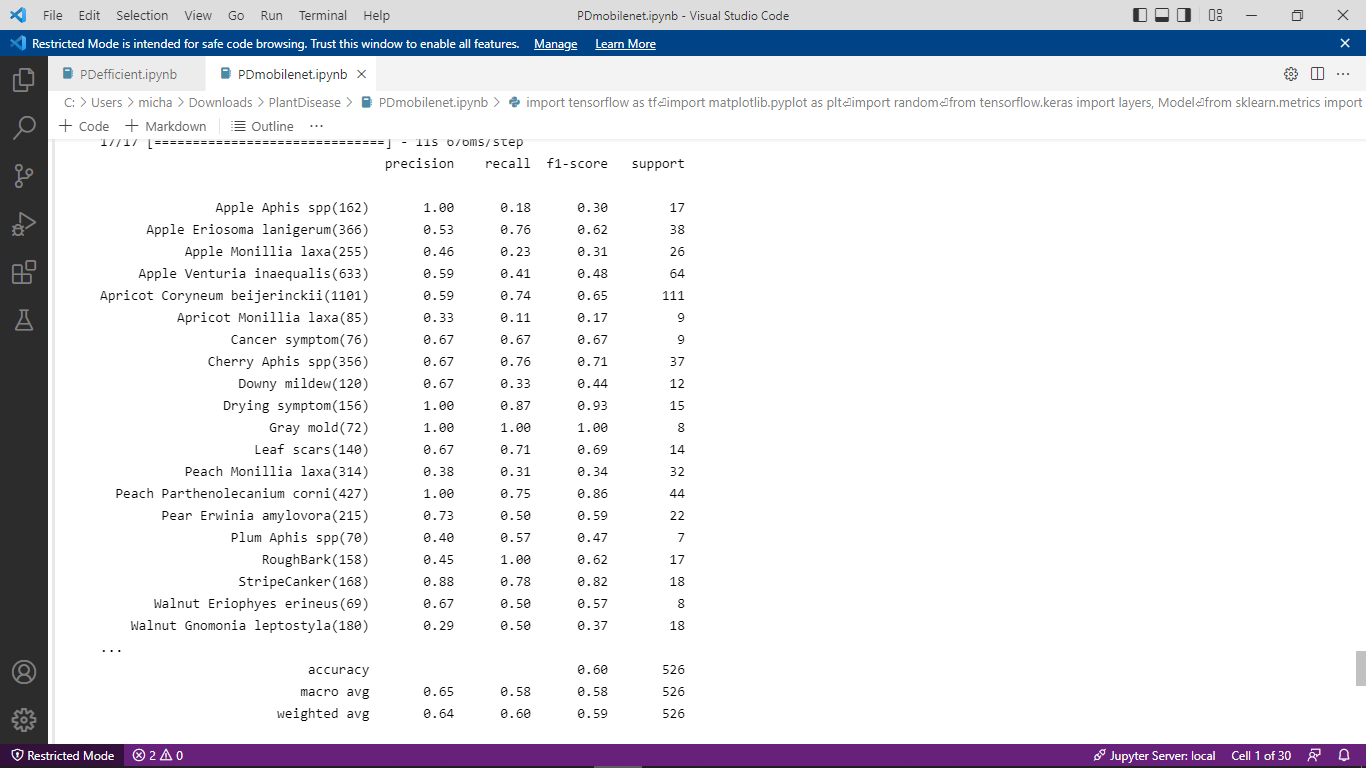
## Hyperparameters

|  |  |
| --- | --- |
| Input data | Image |
| Input Image Shape | 224×224 × 3 |
| Batch Size | 32 |
| Number of epochs | 40 |
| Loss function | Categorical Crossentropy |
| Activation function | ReLU , softmax, swish |
| Learning Rate | 0.001 |
| Optimizer | Adam |
| Dropout rate | 0.2 |

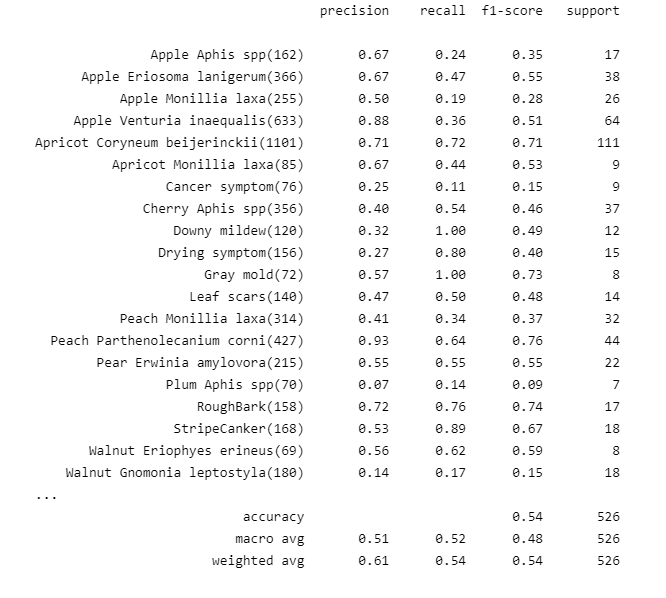
## Proposed Method

| Model | Accuracy (%) | | |
| --- | --- | --- | --- |
| Train | Test | Validation |
| MobileNet | 98.72 | 60.46 | 62.75 |
| CNN | 97.35 | 53.80 | 53.19 |
| VGG19 | 97.03 | 80.99 | 80.48 |
| Resnet50 | 99.41 | 83.27 | 85.46 |
| DenseNet201 | 98.31 | 69.77 | 67.73 |
| **EfficientNet** | **99.51** | **90.49** | **88.25** |

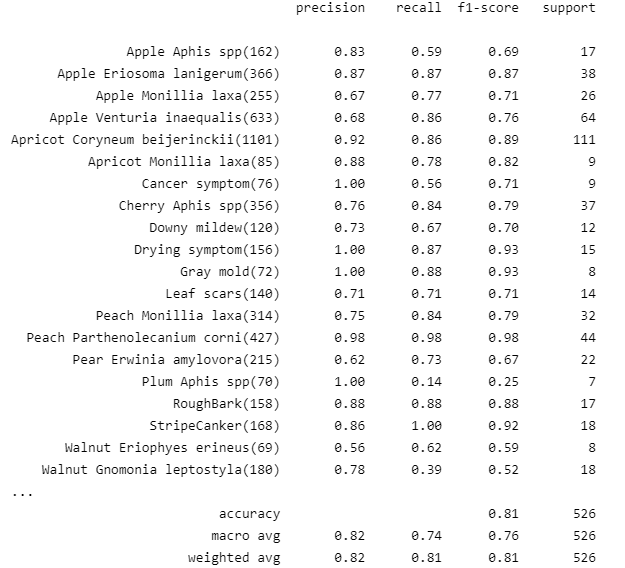
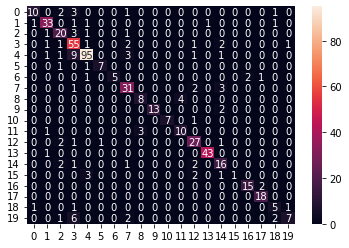
MobileNet :

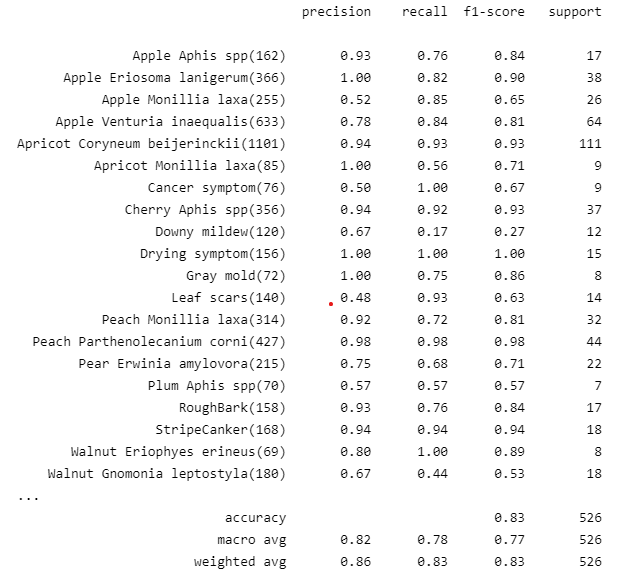
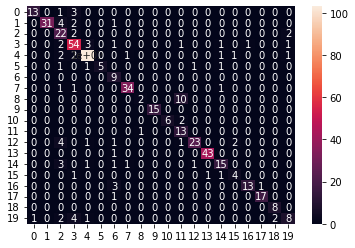
CNN:

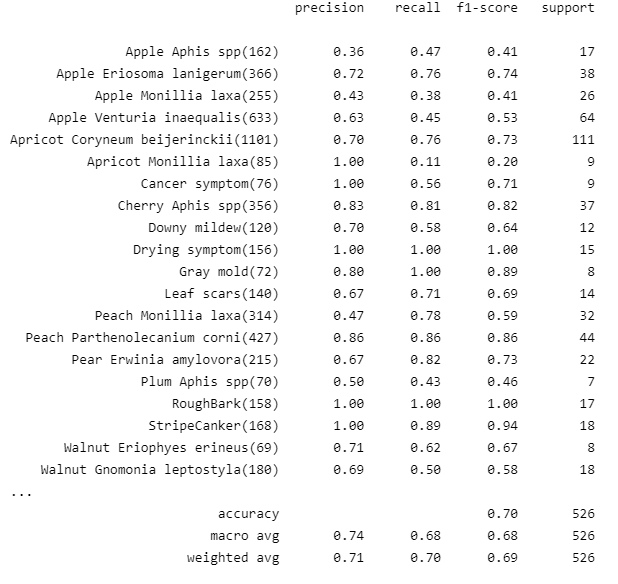
VGG19:

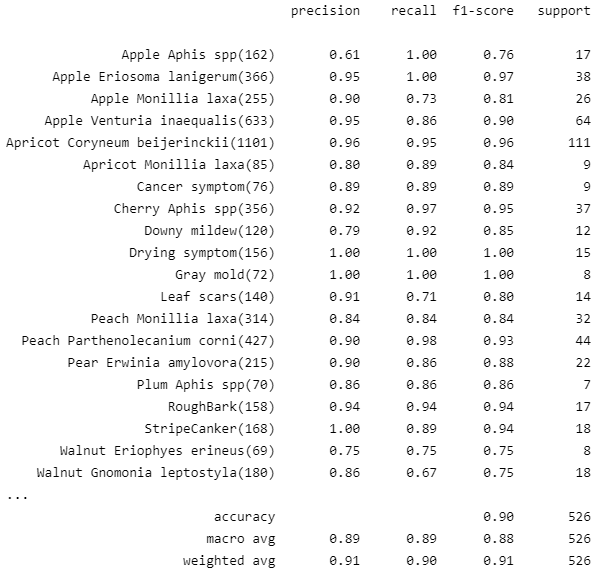
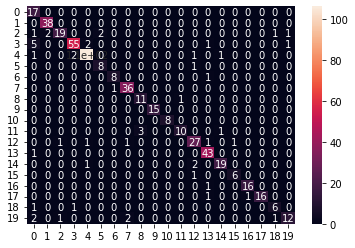
ResNet50:

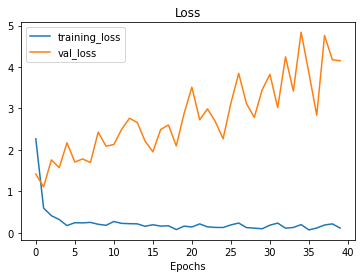
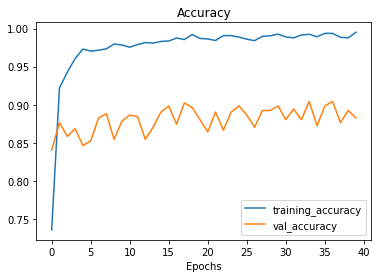
 

DenseNet201:

EfficientNet:

Results on eight different EfficientNet model are :

| Model Number | Accuracy (%) | | |
| --- | --- | --- | --- |
| Train | Test | Validation |
| B0 | 99.51 | 90.49 | 88.25 |
| B1 | 99.39 | 89.16 | 90.24 |
| B2 | 99.22 | 87.64 | 87.85 |
| B3 | 99.02 | 87.83 | 89.64 |
| B4 | 98.90 | 88.21 | 88.84 |
| B5 | 99.02 | 87.83 | 88.45 |
| B6 | 98.23 | 86.12 | 85.86 |
| B7 | 99.09 | 87.83 | 88.45 |

# CONCLUSION

In this study, deep learning models were applied for the diagnosis of plant illnesses using on site images of 20 different classes of plant diseases, based on particular convolutional neural network designs. In this work, EfficientNetB0 is the most successful model architecture and achieved a success rate of 90.49% on previously unseen images (testing accuracy),99.51 training accuracy and 88.25 validation accuracy.

.

##### References

.

1. Turkoglu, M., Yanikoğlu, B. & Hanbay, D. PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *SIViP* **16,**301–309 (2022). <https://doi.org/10.1007/s11760-021-01909-2>
2. Ümit Atila, Murat Uçar, Kemal Akyol, Emine Uçar,Plant leaf disease classification using EfficientNet deep learning model,Ecological Informatics,Volume 61,2021,101182,ISSN 1574-9541, <https://doi.org/10.1016/j.ecoinf.2020.101182>.
3. Ferentinos, Konstantinos P. (2018). *Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145(), 311–318.*doi:10.1016/j.compag.2018.01.009
4. L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
5. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. Plants **2019**, 8, 468. <https://doi.org/10.3390/plants8110468>
6. Jayme Garcia Arnal Barbedo,Plant disease identification from individual lesions and spots using deep learning,Biosystems Engineering,Volume 180,2019,Pages 96-107,ISSN 1537-5110,https://doi.org/10.1016/j.biosystemseng.2019.02.002.