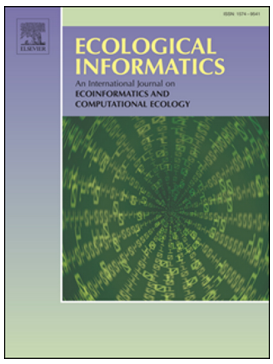


Plant leaf disease classification using EfficientNet deep learning model

Ümit Atila, Murat Uçar, Kemal Akyol, Emine Uçar

PII: S1574-9541(20)30132-1

DOI: https://doi.org/10.1016/j.ecoinf.2020.101182

Reference: ECOINF 101182

To appear in: *Ecological Informatics*

Received date: 3 June 2020

Revised date: 14 September 2020

Accepted date: 2 October 2020

Please cite this article as: Ü. Atila, M. Uçar, K. Akyol, et al., Plant leaf disease classification using EfficientNet deep learning model, *Ecological Informatics* (2019), https://doi.org/10.1016/j.ecoinf.2020.101182

Plant leaf disease classification using EfficientNet deep learning model

Ümit ATİLA[a*], Murat UÇAR[b], Kemal AKYOL[c], Emine UÇAR[b]

[a] Department of Computer Engineering, Faculty of Engineering, Karabuk University, Karabuk, Turkey
[b] Department of Management Information Systems, Faculty of Business and Management Science, Iskenderun Technical University, Hatay, Turkey
[c] Department of Computer Engineering, Faculty of Engineering, Kastamonu University, Kastamonu, Turkey

Abstract: Most plant diseases show visible symptoms, and the technique which is accepted today is that an experienced plant pathologist diagnoses the disease through optical observation of infected plant leaves. The fact that the disease diagnosis process is slow to perform manually and another fact that the success of the diagnosis is proportional to the pathologist's capabilities makes this problem an excellent application area for computer-aided diagnostic systems. Instead of classical machine learning methods, in which manual feature extraction should be flawless to achieve successful results, there is a need for a model that does not need pre-processing and can perform a successful classification. In this study, EfficientNet deep learning architecture was proposed in plant leaf disease classification and the performance of this model was compared with other state-of-the-art deep learning models. The PlantVillage dataset was used to train models. All the models were trained with original and augmented datasets having 55448 and 61486 images respectively. EfficientNet architecture and other deep learning models were trained using transfer learning approach. In the transfer learning, all layers of the models were set to be trainable. The results obtained in the test dataset showed that B5 and B4 models of EfficientNet architecture achieved the highest values compared to other deep learning models in original and augmented datasets with 99.91% and 99.97% respectively for accuracy and 98.42% and 99.39% respectively for precision.

Keywords: Plant Disease, Leaf Image, Deep Learning, Transfer Learning

1 Introduction

Timely and accurate diagnosis of plant diseases is of great importance for sustainable and correct agriculture, as well as for preventing unnecessary waste of financial and other resources. Some plant diseases do not have visible symptoms and it is inevitable to use advanced analysis methods in such diseases. However, the majority of plant diseases show visible symptoms, and the technique that is accepted today is that an experienced plant pathologist diagnoses the disease through optical observation of infected plant leaves (Sankaran et al., 2010; Sladojevic et al., 2016). For a plant pathologist to accurately diagnose plant disease, he must have good observation skills and thus identify characteristic symptoms. However, the excessive variety of plants, variations in the course of plant diseases due to climate changes and the faster spread of diseases to other regions where they have not been seen before, even lead experienced pathologists fail to diagnose certain diseases (Sladojevic et al., 2016). The presence of an expert and intelligent systems that can automatically diagnose plant disease accurately provides valuable contributions to agronomists. On the other hand, offering such a system with a simple mobile application that even non-expert farmers can use is also a good achievement for farmers who do not have an agronomic and phytopathological support infrastructure (Ferentinos, 2018). Advances in artificial intelligence technologies have paved the way for the development of automated systems that can get faster and more accurate results in the diagnosis of diseases. Today, systems that automatically diagnose wide variety of diseases based on artificial intelligence are often used (Jiang et al., 2017). In the last decade, many traditional machine learning models were proposed for the detection and classification of plant diseases. Rumpf et al. studied the early diagnosis and classification of diseases seen in sugar beet based on spectral plant indexes using Support Vector Machine (SVM) (Rumpf et al., 2010). In order to detect five different plant leaf diseases, Al-Hiary et al. performed the segmentation of the diseased areas by clustering the properties obtained with the preprocessing steps using K-Means, and then classified those regions with Artificial Neural Networks (ANN) after performing color and texture based feature extraction (Al-Hiary et al., 2011). Revathi and Hemalatha proposed a method to detect 6 different types of disease seen on cotton leaf. The method they suggest performs feature selection using Particle Swarm Optimization from feature vector including edge, color and texture-based features obtained by image processing and classifies the disease with Cross Information Gain Deep forward Neural Network (Revathi and Hemalatha, 2014). In another study that uses SVM method, Mokhtar et al. performed the detection and identification of two different viruses that show their symptoms on the tomato leaf and cause the disease (Mokhtar et al., 2015). In another study, Pantazi et al. realized the recognition of three different vine leaf diseases by SVM method using the features obtained with the Local Binary Pattern method (Pantazi et al., 2016). In another study, Johannes et al. proposed the use of image processing-based candidate

* Corresponding author: Ümit ATİLA, umitatila@karabuk.edu.tr

hot-spot detection and Naive Bayes classifier for mobile based early diagnosis of three different wheat diseases. They deployed their proposed method on smartphones and evaluated them in a real field environment (Johannes et al., 2017). Recently, Chen et al. proposed a new model named GMDH (Group Method of Data Handling) logistic algorithm to automatically detect plant diseases (Chen et al., 2020a).

The feature extraction process required to perform classification in machine learning is a difficult process and directly affects the classification performance. Increasing capacities and speeds of Central Processing Units (CPUs) and Graphical Processing Units (GPUs) paved the way for the development of new high-performance methods that can process raw data without the need for handcrafted features, and this led to deep learning architectures (LeCun et al., 2015). Deep neural network architectures with many processing layers and neurons can efficiently perform high-complexity tasks such as voice and image recognition by processing large-size data. The use of deep learning methods in the diagnosis and classification of diseases from medical images is quite common (Shen et al., 2017). On the other hand, in a review article of 2019 (Saleem et al., 2019), deep learning-based studies for the detection and classification of plant leaf diseases were examined and the potentials of deep learning were evaluated. It has been observed that most studies in the literature use the PlantVillage dataset and diagnose disease for a particular plant or several plants rather than classifying all plant diseases in this dataset. In one of these studies, Sladojevic et al. performed the classification of 13 different plant diseases using Convolutional Neural Network (CNN) (Sladojevic et al., 2016). In their study, they used 30880 images to train their proposed model and 2589 images to test. Their proposed model achieved an average of 96.3% accuracy. Chen et al. performed the detection of rice plant disease with a deep transfer learning-based model called DENS-INCEP (Chen et al., 2020c). In another study, Chen et al. performed maize plant and rice plant disease classification by modifying the VGGNet module (Chen et al., 2020d). It has been observed that there are five studies that classify all diseases in the PlantVillage dataset consisting of 39 classes in total including 38 different plant diseases and a class for background images without leaf. Mohanty et al., classified plant diseases using CNN models such as AlexNet and GoogLeNet (Mohanty et al., 2016). In their study, they obtained 99.35% classification accuracy. Too et al., used CNN models such as VGG16, Inception V4, ResNet50, ResNet101, Resnet152 and DenseNets 121. It was reported that the DenseNet architecture used in the study had fewer parameters and low calculation time compared to other models and gave the highest test accuracy with 99.75% (Too et al., 2019). Geetharamani and Pandian trained the 9-layer CNN architecture in the PlantVillage dataset with different epoch, batch size and dropout and compared the performance of achieved models with popular transfer learning approaches. Their proposed model achieved 96.46% classification accuracy on the test dataset (Geetharamani and Pandian, 2019). On the other hand, there are two more studies on other versions of the PlantVillage dataset that have been expanded with extra images. In one of these, Ferentinos made the classification of 58 different diseases of 25 different plant species using 87848 images with AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat and VGG architectures. VGG architecture used in the study gave the highest accuracy with 99.53% (Ferentinos, 2018). In the second study, Arsenovic et al. created PlantDisease dataset, which is an expanded version of the PlantVillage dataset and contains 79265 images. They conducted experimental studies using both datasets. While the two-stage PlantDiseaseNet model they proposed for classification determines the plant species from leaves in the first stage, it classifies these leaves in the second stage. The model they proposed achieved 93.67% accuracy in the Plant Disease dataset (Arsenovic et al., 2019). Some studies have also been conducted to investigate the performance of deep learning architectures in plant disease classification in both PlantVillage and private datasets. Nanehkaran et al. proposed a new model for the detection of plant diseases, including image segmentation and image classification stages (Nanehkaran et al., 2020). They proposed a hue, saturation and intensity-based and LAB-based hybrid segmentation algorithm in the image segmentation phase and used CNN model in the classification phase. Chen et al. proposed a new model for the detection of plant diseases called MobileNet-Beta by expanding the pre-trained MobileNetV2 model with the Classification Activation Map (Chen et al., 2020b). They tested the proposed model on the PlantVillage dataset and on their own dataset. According to the test results, MobileNet-Beta model achieved 99.85% accuracy in the PlantVillage dataset and 99.11% accuracy on their own dataset.

As seen from previous studies mentioned above, there is an increase in the use of deep learning architectures on the diagnosis of plant leaf diseases in the literature. However, there are still gaps to be investigated regarding the use of especially new deep learning architectures in plant leaf disease detection. Especially, the need for efficient models with fewer parameters, trained faster and without compromise on performance is inevitable.

This study proposes EfficientNet (Tan and Le, 2019) deep learning architecture for the classification of plant diseases. The performance of the proposed model is compared with state-of-the-art CNN architectures such as AlexNet, ResNet50, VGG16 and Inception V3.

The rest of this study is organized as follows. Section 2 describes the dataset and deep neural network architectures used in this study. Experimental studies are given in Section 3. The results obtained in the study are given and discussed in Section 4. The study is concluded with Section 5.

2 Materials and Methods

2.1 Dataset

In this study, PlantVillage dataset is used containing 38 classes and 54305 images of 14 different plant species in total, 12 of which are healthy, 26 of which are diseased (Hughes and Salathe, 2015). Images in the dataset are colored images of varying sizes. The dataset also has one more class identifying 1143 background images. Thus, the total number of images in the dataset is 55448. Fig. 1 shows 8 different plant-disease pairs selected randomly.

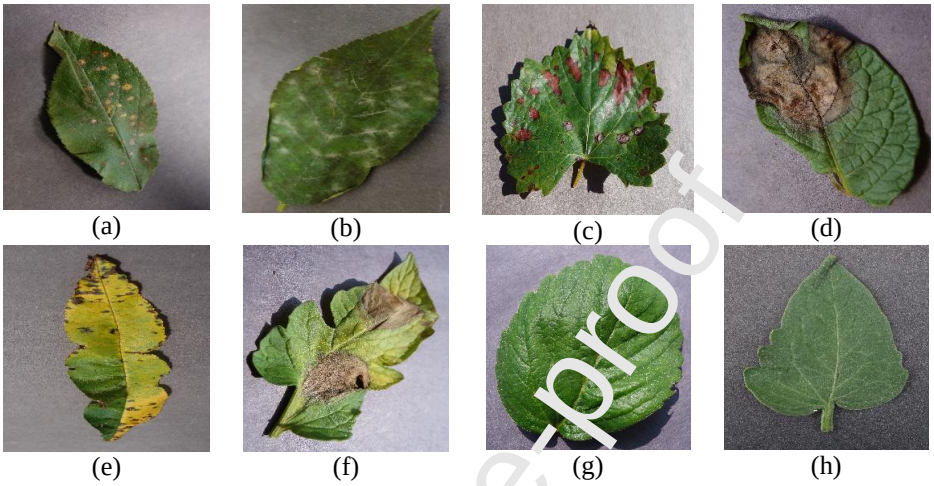


Figure 1. Sample plant-disease pairs from PlantVillage dataset: a) apple with cedar apple rust b) cherry with powdery mildew c) grape with black measles d) potato with late blight e) peach with bacterial spot f) tomato with late blight g) healthy strawberry h) healthy tomato

Geetharamani and Pandian (Geetharamani and Pandian, 2019) applied six different augmentation methods to the PlantVillage dataset in their study, and they shared the original and augmented datasets for use in scientific studies at https://data.mendeley.com/datasets/tywbtsjrjv/1. In the study, the number of samples belonging to classes with less than 1000 samples were increased to 1000 with augmentation methods, and no action was taken on the images in other classes. After applying augmentation, the number of images in the dataset has been increased to 61486. In this study, we use the original dataset as well as the augmented dataset by Geetharamani and Pandian to increase the success of deep neural networks.

2.2 State of the art CNN based models

The performance of the EfficientNet architecture proposed in this study is compared with state-of-the-art CNN architectures such as AlexNet, ResNet50, VGG16 and Inception V3.

2.2.1 AlexNet

The AlexNet model (Krizhevsky et al., 2012) is an 8-layer CNN architecture with approximately 61 million parameters introduced in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) competition held in 2012. This architecture follows the design of LeNet-5 (LeCun et al., 1989). The AlexNet architecture given in Fig. 2 takes the input image size as 227x227 and consists of 5 convolutional layers followed by 3 fully connected layers and finally the Softmax layer. This architecture uses ReLU activation function in the convolutional and fully connected layers. The fully connected layer FC-8 of AlexNet architecture used in this study is connected to the Softmax layer with 39 neurons. Each output value in the Softmax layer is the ratio of the input image to the class represented by the corresponding output. To this aim, Softmax layer generates a distribution using the inputs coming from the FC-8 layer and assigns a probability value for each class which of total is 1 for all classes.

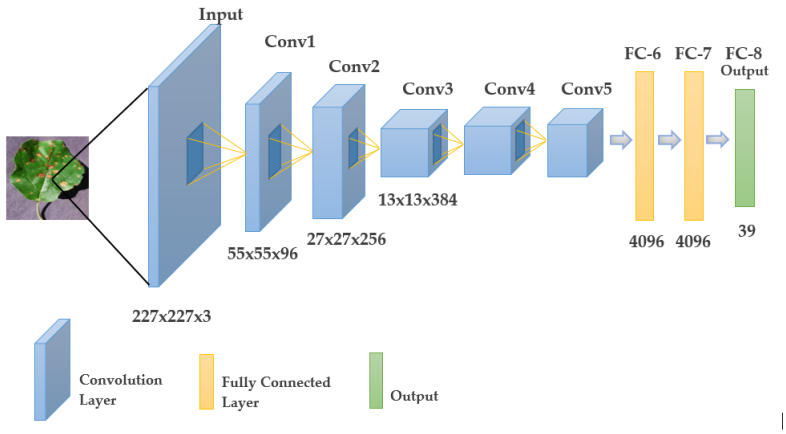


Figure 2. Schematic representation of AlexNet

2.2.2 VGG16

VGG16 (Simonyan and Zisserman, 2014), which won the ILSVRC-2014, is a CNN architecture with approximately 138 million parameters which won ILSVRC-2014. The most striking feature of this architecture is that instead of having large number of hyper-parameters, it always has the same convolution layers that use 3x3 filters with stride 1 and same padding and maximum pooling layers that use 2x2 filters with stride 2. The VGG16 architecture follows this convolution and maximum pooling layers arrangement consistently throughout the entire architecture (Fig. 3). Finally, there are 3 FC layers first two with ReLU and the last with Softmax activation function. This architecture contains 16 layers and the input layer takes images of 224x224 pixels.

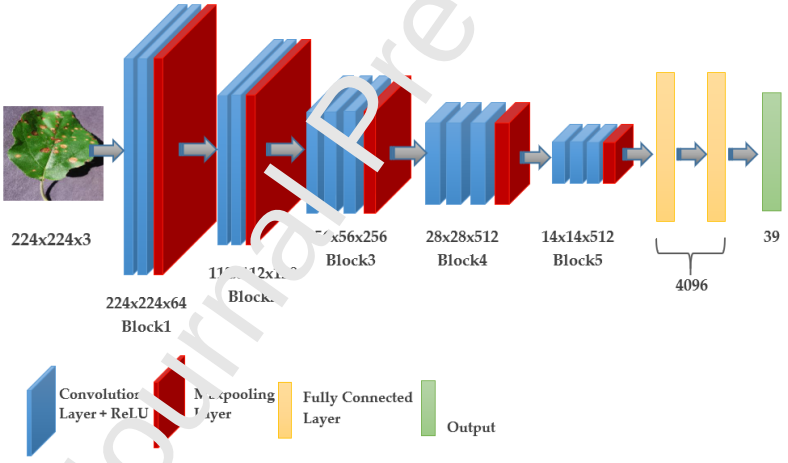


Figure 3. Schematic representation of VGG16

2.2.3 ResNet50

ResNet50 architecture (He et al., 2016), which won the ILSVRC-2015 competition in 2015, is an architecture proposed to solve the problem of multiple non-linear layers not learning identity maps and degradation problem. ResNet50 is a network in network architecture based on many stacked residual units (Fig. 4). Residual units are used as building blocks to build the network. These units consist of convolution and pooling layers. This architecture uses 3x3 filters as VGG16 and takes input images of 224x224 pixels.

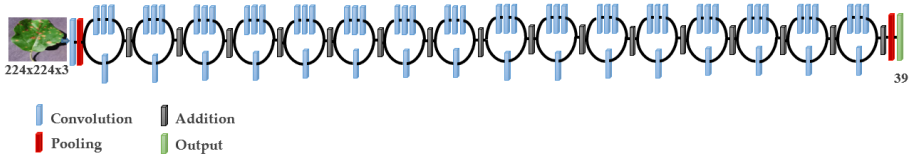


Figure 4. Schematic representation of ResNet50

2.2.4 Inception V3

Inception V3 (Szegedy et al., 2016) developed by Google, is the third release in the Deep Learning Evolutionary Architectures series. After the Inception V1 architecture was developed by Szegedy, batch normalization was performed in Inception V2. Then the idea of factorization was introduced in Inception V3. The main purpose in factorization is to reduce the number of connections and parameters without reducing the efficiency of the network. The model itself consists of symmetrical and asymmetrical building blocks containing convolutions, average pooling, max pooling, concats, dropouts and fully connected layers (Fig. 5). The Inception V3 architecture, which has the Softmax function in the last layer, consists of 42 layers in total and the input layer takes images of 299x299 pixels.

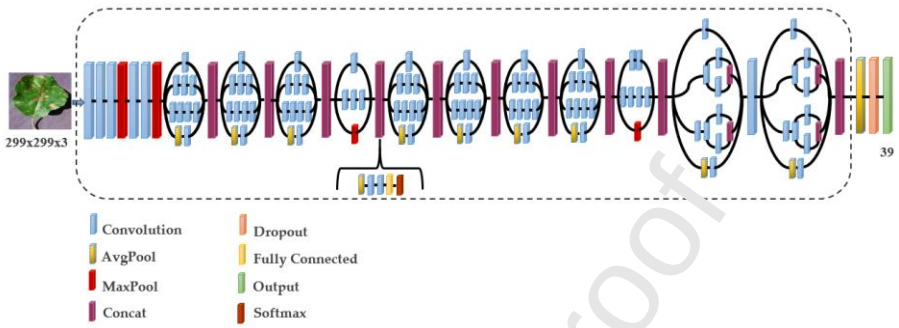


Figure 5. Schematic representation of Inception V3

2.2.5 EfficientNet

Success has increased as the models used in the ImageNet dataset since 2012 have become more complex, but many are not effective in terms of computing load. EfficientNet model, which is among the state-of-the-art models by reaching 84.4% accuracy with 66M parameter in the ImageNet classification problem, can be considered as a group of CNN models. EfficientNet group consists of 8 models between B0 and B7, and as the model number grows, the number of calculated parameters does not increase much, while accuracy increases noticeably. Unlike other CNN models, EfficientNet uses a new activation function called Swish instead of the Rectifier Linear Unit (ReLU) activation function (Tan and Le, 2019).

The aim of deep learning architectures is to reveal more efficient approaches with smaller models. EfficientNet, unlike other state-of-the-art models, achieves more efficient results by uniformly scaling depth, width, and resolution while scaling down the model. The first step in the compound scaling method is to search for a grid to find the relationship between the different scaling dimensions of the baseline network under a fixed resource constraint. In this way, a suitable scaling factor for depth, width and resolution dimensions is determined. These coefficients are then applied to scale the baseline network to the desired target network (Tan and Le, 2019).

The main building block for EfficientNet is the inverted bottleneck MBConv, which was first introduced in MobileNetV2 (Mark et al., 2018), but due to the increased FLOPS (floating point operations per second) budget, it is used slightly more than MobileNetV2. In MBConv, blocks consist of a layer that first expands and then compresses the channels, so direct connections are used between bottlenecks that connect much fewer channels than expansion layers. This architecture has in-depth separable convolutions that reduce calculation by almost k^2 factor compared to traditional layers where k is the kernel size which denotes the width and height of the 2D convolution window (Mark et al., 2018). The schematic representation of EfficientNet B0 model is shown in Fig. 6.

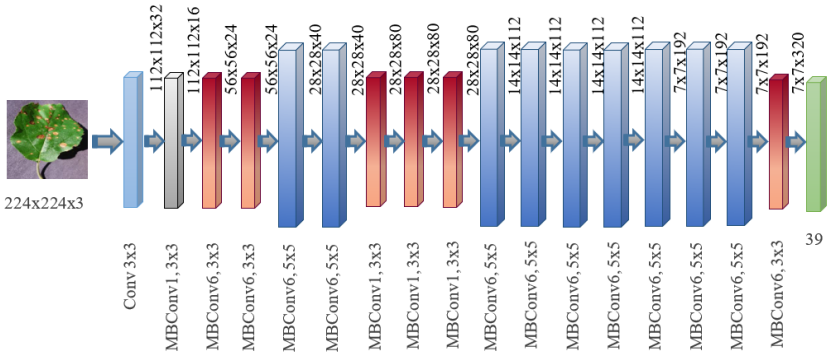


Figure 6. Schematic representation of EfficientNet

In compound scaling, the compound coefficient φ is used with the principles given in Eq. 1 to uniformly scale depth, width, and resolution.

$$\begin{aligned} &depth: d = \alpha^{\varphi} \\ &width: w = \beta^{\varphi} \\ &resolution: r = \gamma^{\varphi} \end{aligned}$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \qquad (1)$$

where, α, β, γ are constants that can be determined by grid search. φ is a user-defined coefficient that controls how many resources are available for model scaling, while α, β, γ determine how these extra resources are assigned to the network width, depth and resolution, respectively. In a regular convolution process, FLOPS are proportional to d, w^2, r^2. Since the cost of computing in convolution networks is largely due to convolution operations, scaling convolution network as given in Eq. 1 increases the FLOPS of the network by approximately $(\alpha, \beta^2, \gamma^2)^{\varphi}$ in total (Tan and Le, 2019).

Starting from Baseline EfficientNet-B0, the compound scaling method scales this model in two steps (Tan and Le, 2019):

Step 1: Assuming that there are twice as many resources available, grid search is performed with $\varphi=1$ and the best values are found for α, β, γ.

Step 2: Obtained α, β, γ values are determined as constant and the baseline network is scaled up to obtain EfficientNet-B1 to B7 using Eq.1 with different φ values.

2.2.6 Transfer Learning

Transfer learning is a machine learning technique in which the knowledge gained during training in a problem is used for training in another task or field (Weiss et al., 2016). In deep learning, the first few layers are trained to define the characteristics of the task. During transfer learning, the last few layers of the trained network can be removed and retrained with new layers for the target task. In the transfer learning approach, using the knowledge of the network previously trained with large amounts of visual data in a new task is very advantageous in terms of saving time and achieving high accuracy compared to training a model from scratch.

3 Experiments

3.1 Experimental Setup

All models used in this study were compiled with GPU support. All experimental studies were conducted in Google cloud environment on a 64-bit Debian GNU / Linux 9.11 operating system running on Intel (R) Xeon (R) Gold CPU @ 2.20GHz CPU and 16 GB RAM with NVIDIA Tesla K80 having 12GB memory. All codes are realized with Keras 2.3.1 framework, which is an open source deep neural network library written in Python language.

3.2 Training

The original and augmented PlantVillage datasets used in this study were randomly divided into training, validation and test sets by 90%, 7% and 3%, respectively. Training and validation datasets were only used for training and fitting of the model, while the test set was used to evaluate the prediction performance on samples that the model did not see before. Table 1 summarizes the number of images used in each dataset according to these rates.

Table 1. Division of images into train, test and validation sets

	Total	**Train (90%)**	**Validation (7%)**	**Test (3%)**
Original dataset	55448	49903	3813	1663
Augmented dataset	61486	55337	4304	1844

Existing CNN models in this study were used with the transfer learning approach. To speed up learning, CNN models were fine-tuned to identify and classify all categories in the dataset, with pre-trained models on ImageNet dataset. The ImageNet dataset contains approximately 1.2 million images and 1000 class categories. Pre-trained weights of the state-of-the-art CNN models on ImageNet were used to shorten the training time of the deep learning models used in this study. The last FC layers of all models with 1000 outputs were changed to 39 outputs in accordance with the problem. All layers of pre-trained models were set as trainable. Softmax was

chosen as the activation function in the last layer and loss function were selected as categorical cross entropy. During the training, the early stopping technique was used with patience as 5 and minimum change in loss as 1e-3. Since early stopping technique was used, maximum epoch was not defined in the training of models.

Pre-trained models were used with the same optimization method as used on training of ImageNet dataset. Accordingly, while VGG16 model uses SGD optimization method, all other models use Adam optimization method. In addition, while the learning rate was selected as 0.001 for Adam method, it was set to 0.01 for SGD method. Validation step for all models was set to 1.

In this study, each pixel value of images in the original and augmented datasets was first normalized dividing by 255. The images were then resized to the default size accepted by each model. Accordingly, images were set to 227x227 pixels for AlexNet, 224x224 pixels for ResNet50 and VGG16, and 299x299 pixels for Inception V3. In our experimental studies, input image resolutions were necessarily resized for all models of EfficientNet architecture due to our hardware limitations. Table 2 summarizes the default image resolutions and number of parameters defined for EfficientNet models. In the trial and error studies, it was seen that the maximum allowed input size that our hardware resources were sufficient for the training of the B7 model which has the highest number of parameters was 132x132. Therefore, the input size for all models of EfficientNet architecture was set as 132x132 in order to evaluate all models under same conditions.

Table 2. The default input image sizes and number of total parameters for EfficientNet models

Model Number	Input size	Number of total parameters
B0	224x224	5,330,571
B1	240x240	7,856,239
B2	260x260	9,177,569
B3	300x300	12,320,535
B4	380x380	19,466,823
B5	456x456	30,562,527
B6	528x528	43,265,143
B7	600x600	66,658,687

Mini-batch is the size of the data used to update weights and bias during backpropagation. The mini batch value is chosen smaller than the total dataset size and generally a value that can divide the total number of samples in the dataset is preferred. This value contributes to learning processes by balancing network convergence rate and accurate prediction (Li et al., 2014). In this study, mini-batch size was set to 16, which was the maximum value allowed by hardware resources in all models. Table 3 summarizes the main parameters used in all experiments.

Table 3. The parameter values used for deep learning models

Model name	Image size	Optimization method & parameters	Learning rate
AlexNet	227x227	Adam (Beta 1 = 0.9, Beta 2= 0.999, Decay=0.0)	0.001
VGG16	224x224	SGD (Momentum=0.0, Decay=0.0)	0.01
ResNet50	224x224	Adam (Beta 1 = 0.9, Beta 2= 0.999, Decay=0.0)	0.001
Inception V3	299x299	Adam (Beta 1 = 0.9, Beta 2= 0.999, Decay=0.0)	0.001
EfficientNet	132x132	Adam (Beta 1 = 0.9, Beta 2= 0.999, Decay=0.0)	0.001

3.3 Performance Metrics

Since there are 39 categories in the PlantVillage dataset, multi-class classification was performed. Considering the values in the confusion matrix obtained in such classifications, the metrics given between Eq. 2-9 are calculated using indices such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Here, TP is the number of correctly classified diseased images in each category, while TN, on the other hand, represents the sum of the correctly classified images in all other categories except for the relevant category. FN gives the number of misclassified images from the relevant category. FP gives the number of misclassified images in all other categories except for the relevant category.

The performances of EfficientNet and other state-of-the-art CNN models discussed in this study are measured using different metrics such as Accuracy (Acc), Sensitivity (Sen), Specificity (Spe) and Precision (Pre). Sensitivity is the ratio of correctly predicted positives out of all true positives. Specificity is the ratio of correctly predicted negatives out of all true negatives. On the other hand, accuracy indicates the rate of correctly classified samples out of all samples. Precision is the proportion of correctly predicted positives out of all positive identifications. These metrics and their extended calculations for multi-class classification using macro-averaging (Sokolova and Lapalme, 2009) are given in between Eq. 2-9.

For a class *k*,

$$Sen(k) = \frac{\#TP(k)}{\#TP(k)+\#FN(k)} \quad (2)$$

$$Spe(k) = \frac{\#TN(k)}{\#TN(k)+\#FP(k)} \quad (3)$$

$$Acc(k) = \frac{\#TP(k)+\#TN(k)}{\#TP(k)+\#FN(k)+\#TN(k)+\#FP(k)} \quad (4)$$

$$Pre(k) = \frac{\#TP(k)}{\#TP(k)+\#FP(k)} \quad (5)$$

$$AverageSen = \frac{1}{\#classes}\sum_{k=1}^{\#classes} Sen(k) \quad (6)$$

$$AverageSpe = \frac{1}{\#classes}\sum_{k=1}^{\#classes} Spe(k) \quad (7)$$

$$AverageAcc = \frac{1}{\#classes}\sum_{k=1}^{\#classes} Acc(k) \quad (8)$$

$$AveragePre = \frac{1}{\#classes}\sum_{k=1}^{\#classes} Pre(k) \quad (9)$$

4 Results and Discussions

The main purpose of this study is to examine the success of EfficientNet deep learning architecture in the classification of plant leaf disease and to compare with the performances of state-of-the-art CNN models in the literature. As mentioned in Section 3.2, all deep learning models used in this study were trained by performing transfer learning.

All experimental studies were carried out on both original and augmented datasets. In this context, the average accuracy, sensitivity, specificity and precision values obtained by all models on the test dataset were given in Table 4 and Table 5, respectively, for the original and augmented datasets. Since early stopping was used during the training, the period until the epoch where the loss values of the models started to increase was evaluated as the total training time. The time per epoch in the training of models was calculated by dividing the total training time by the total number of epochs and given in Table 4 and Table 5. Values marked in bold in Table 4 and Table 5 represent the case where best value obtained for the relevant performance criterion.

As seen in Table 4, in the original dataset, all models obtained average accuracy values very close each other. In the original dataset, EfficientNet B5 model gave the best average sensitivity value for all classes. On the other hand, the average successes of models in predicting the classes other than the target class were close to each other. The true classification rate of the samples that B5 model classifies as positive (precision) was higher than the other models. While the B5 model completed 1 epoch in 1930 seconds, the training was completed in 643.3 minutes. The lowest training time per epoch achieved by AlexNet.

Table 4. Average results of deep learning models on original dataset

Model	Avg Acc (%)	Avg Sen (%)	Avg Spe (%)	Avg Pre (%)	Time per epoch (sec)	Total training time (min)
B0	99.81	96.26	99.90	96.85	552	64.4
B1	99.79	96.00	99.89	96.61	800	80.0
B2	99.83	96.72	99.91	97.40	839	69.9
B3	99.83	96.77	99.91	97.20	1065	177.5
B4	99.84	96.82	99.92	97.24	1405	210.8
B5	**99.91**	**98.31**	**99.96**	**98.42**	1930	643.3
B6	99.69	94.00	99.84	94.99	2405	80.2
B7	99.86	97.23	99.93	97.60	3240	918.0
AlexNet	99.45	89.33	99.72	90.26	3[illegible]0	82.7
ResNet50	99.77	95.44	99.88	96.53	174[illegible]	203.0
VGG16	99.81	96.26	99.90	96.83	146[illegible]5	146.5
Inception V3	99.81	96.26	99.90	96.84	2180	399.7

As seen in Table 5, the evaluations expressed for the B5 model in Table 4 for the original dataset can be said for the B4 model in augmented dataset. While the B4 model completed 1 epoch in 1548 seconds during the training, the training was completed in 283.8 minutes. The lowest training time per epoch achieved by AlexNet.
As can be seen from both tables, B4 and B5 models offered very close performance, but we can say that the B4 model was slightly more successful than the B5 on the augmented dataset.

Table 5. Average results of deep learning models on augmented dataset

Model	Avg Acc (%)	Avg Sen (%)	Avg Spe (%)	Avg Pre (%)	Time per epoch (sec)	Total training time (min)
B0	99.88	97.89	99.94	97.78	617	82.3
B1	99.92	98.46	99.96	98.50	880	176.0
B2	99.91	98.26	99.95	98.37	930	77.5
B3	99.94	98.82	99.97	98.85	1180	157.3
B4	**99.97**	**99.38**	**99.98**	**99.39**	1548	283.8
B5	99.93	98.72	99.97	98.76	2150	179.2
B6	99.93	98.67	99.96	98.73	2685	313.3
B7	99.90	98.00	99.95	98.10	3625	181.3
AlexNet	99.67	93.64	99.83	93.87	352	88.0
ResNet50	99.88	97.59	99.94	97.91	1916	223.5
VGG16	99.94	98.77	99.97	98.82	1622	108.1
Inception V3	99.93	98.62	99.96	98.70	2330	233.0

Fig. 7 and Fig. 8 show the accuracies obtained in the test sets of original and augmented datasets for all models in ascending order. Test accuracy given in these figures was calculated as the ratio of number of correctly classified samples to number of all samples. The B5 model achieved the highest accuracy with 99.91% in the original dataset, while the B4 model achieved the highest accuracy with 99.97% in the augmented dataset. In addition, for both original and augmented datasets, AlexNet gave the lowest accuracy values with 99.45% and 99.67%, respectively. As can be seen from the test accuracy curves, the success of all models on the augmented dataset is higher than the original dataset.

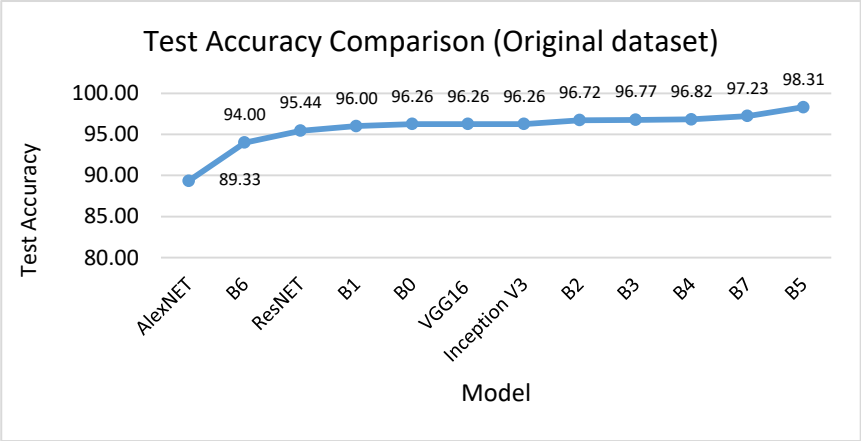


Figure 7. Test accuracies of deep learning models for original dataset

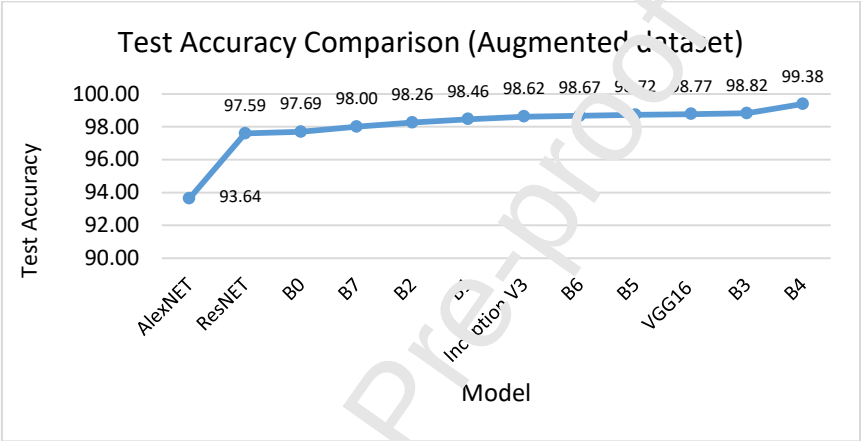


Figure 8. Test accuracies of deep learning models for augmented dataset

The TP, TN, FP, FN, accuracy, sensitivity, specificity and precision values obtained by B5 and B4 models for each class in the original and augmented datasets, respectively, are presented in Table 6 and Table 7, respectively. Considering the precision values of EfficientNet architecture for each class on the test dataset, the B5 model achieved the best performance in the original dataset ranged from 84.75% to 100%, while the B4 model achieved the best performance in the augmented dataset ranged from 96.08% to 100%. On the other hand, when the accuracy values obtained on the test dataset for each class were examined, the B5 model achieved accuracy rate between 99.54% and 100% in the original dataset, while the B4 model achieved accuracy between 99.85% and 100% in the augmented dataset. In addition, confusion matrices of B5 and B4 models were given in Appendix A and Appendix B.

Table 6. Classification performance of B5 on original dataset for each class

Class	TP	TN	FP	FN	Acc (%)	Sen (%)	Spe (%)	Pre (%)
Apple scab	50	1900	0	0	100.00	100.00	100.00	100.00
Apple Black rot	50	1900	0	0	100.00	100.00	100.00	100.00
Apple Cedar apple rust	49	1900	0	1	99.95	98.00	100.00	100.00
Apple healthy	50	1899	1	0	99.95	100.00	99.95	98.04
Background without leaves	50	1900	0	0	100.00	100.00	100.00	100.00
Blueberry healthy	50	1899	1	0	99.95	100.00	99.95	98.04
Cherry Powdery mildew	50	1900	0	0	100.00	100.00	100.00	100.00
Cherry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Corn Cercospora leaf spot Gray leaf spot	46	1899	1	4	99.74	92.00	99.95	97.87
Corn Common rust	50	1900	0	0	100.00	100.00	100.00	100.00
Corn Northern Leaf Blight	49	1896	4	1	99.74	98.00	99.79	92.45
Corn healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Grape Black rot	49	1900	0	1	99.95	98.00	100.00	100.00
Grape Esca (Black Measles)	50	1899	1	0	99.95	100.00	99.95	98.04
Grape Leaf blight (Isariopsis Leaf Spot)	50	1900	0	0	100.00	100.00	100.00	100.00
Grape healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Orange Haunglongbing (Citrus greening)	50	1900	0	0	100.00	100.00	100.00	100.00
Peach Bacterial spot	50	1895	5	0	99.74	100.00	99.74	90.91
Peach healthy	47	1900	0	3	99.85	94.00	100.00	100.00
Pepper, bell Bacterial spot	50	1899	1	0	99.95	100.00	99.95	98.04
Pepper, bell healthy	49	1900	0	1	99.95	98.00	100.00	100.00
Potato Early blight	50	1900	0	0	100.00	100.00	100.00	100.00
Potato Late blight	50	1891	9	0	99.54	100.00	99.53	84.75
Potato healthy	41	1900	0	9	99.54	82.00	100.00	100.00
Raspberry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Soybean healthy	50	1898	2	0	99.90	100.00	99.89	96.15
Squash Powdery mildew	50	1900	0	0	100.00	100.00	100.00	100.00
Strawberry Leaf scorch	50	1900	0	0	100.00	100.00	100.00	100.00
Strawberry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato Bacterial spot	50	1897	3	0	99.85	100.00	99.84	94.34
Tomato Early blight	49	1899	1	1	99.90	98.00	99.95	98.00
Tomato Late blight	47	1899	1	3	99.79	94.00	99.95	97.92
Tomato Leaf Mold	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato Septoria leaf spot	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato Spider mites Two-spotted spider mite	46	1900	0	4	99.79	92.00	100.00	100.00
Tomato Target Spot	48	1899	1	2	99.85	96.00	99.95	97.96
Tomato Yellow Leaf Curl Virus	48	1899	1	2	99.85	96.00	99.95	97.96
Tomato mosaic virus	49	1900	0	1	99.95	98.00	100.00	100.00
Tomato healthy	50	1899	1	0	99.95	100.00	99.95	98.04

Table 7. Classification performance of B4 on augmented dataset for each class

Class	TP	TN	FP	FN	Acc (%)	Sen (%)	Spe (%)	Pre (%)
Apple scab	50	1900	0	0	100.00	100.00	100.00	100.00
Apple Black rot	50	1900	0	0	100.00	100.00	100.00	100.00
Apple Cedar apple rust	50	1900	0	0	100.00	100.00	100.00	100.00
Apple healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Background without leaves	50	1898	2	0	99.90	100.00	99.89	96.15
Blueberry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Cherry Powdery mildew	50	1900	0	0	100.00	100.00	100.00	100.00
Cherry healthy	49	1900	0	1	99.95	98.00	100.00	100.00
Corn Cercospora leaf spot Gray leaf spot	49	1898	2	1	99.85	98.00	99.89	96.08
Corn Common rust	50	1900	0	0	100.00	100.00	100.00	100.00
Corn Northern Leaf Blight	48	1899	1	2	99.85	96.00	99.95	97.96
Corn healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Grape Black rot	50	1899	1	0	99.95	100.00	99.95	98.04
Grape Esca (Black Measles)	49	1900	0	1	99.95	98.00	100.00	100.00
Grape Leaf blight (Isariopsis Leaf Spot)	50	1900	0	0	100.00	100.00	100.00	100.00
Grape healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Orange Haunglongbing (Citrus greening)	50	1900	0	0	100.00	100.00	100.00	100.00
Peach Bacterial spot	50	1900	0	0	100.00	100.00	100.00	100.00
Peach healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Pepper, bell Bacterial spot	50	1900	0	0	100.00	100.00	100.00	100.00
Pepper, bell healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Potato Early blight	50	1900	0	0	100.00	100.00	100.00	100.00
Potato Late blight	50	1899	1	0	99.95	100.00	99.95	98.04
Potato healthy	50	1899	1	0	99.95	100.00	99.95	98.04
Raspberry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Soybean healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Squash Powdery mildew	50	1900	0	0	100.00	100.00	100.00	100.00
Strawberry Leaf scorch	50	1900	0	0	100.00	100.00	100.00	100.00
Strawberry healthy	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato Bacterial spot	48	1900	0	2	99.90	96.00	100.00	100.00
Tomato Early blight	49	1899	1	1	99.90	98.00	99.95	98.00
Tomato Late blight	49	1899	1	1	99.90	98.00	99.95	98.00
Tomato Leaf Mold	48	1900	0	2	99.90	96.00	100.00	100.00
Tomato Septoria leaf spot	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato Spider mites Two-spotted spider mite	50	1899	1	0	99.95	100.00	99.95	98.04
Tomato Target Spot	49	1900	0	1	99.95	98.00	100.00	100.00
Tomato Yellow Leaf Curl Virus	50	1899	1	0	99.95	100.00	99.95	98.04
Tomato mosaic virus	50	1900	0	0	100.00	100.00	100.00	100.00
Tomato healthy	50	1900	0	0	100.00	100.00	100.00	100.00

For the B5 and B4 models, which give the best performances in the original and augmented datasets, the changes in the validation accuracy by epoch were shown in Fig. 9a and Fig. 9b respectively. The early stopping technique used in the training determined the 20^{th} and 11^{th} epochs as the restoring points for which the validation loss started to increase for the B5 and B4 models respectively (Fig. 9c and Fig. 9d).

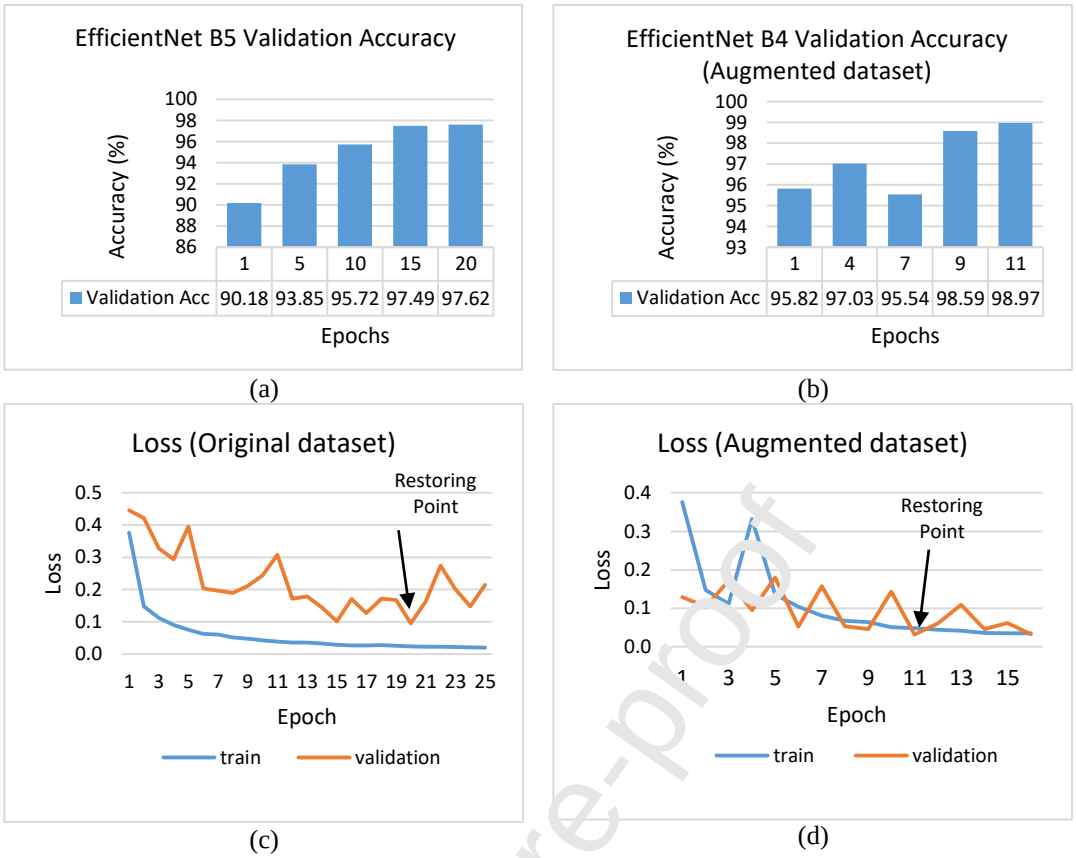


Figure 9. a-b) Validation accuracies for different epochs on original and augmented datasets, c-d) Loss curves for different epochs on training and validation sets of original and augmented datasets

The accuracy and loss curves obtained for the B5 and B4 models on training and validation sets were given in Fig. 10a and Fig. 10b, respectively. As seen in Fig. 10a, the training accuracy in the original dataset exceeded 98% in 6 epochs and the loss was significantly reduced at the end of 20 epochs, resulting in the highest accuracy. As seen in Fig. 10b, in the augmented dataset, the loss was significantly reduced at the end of 11 epochs and the highest accuracy was achieved over 98%.

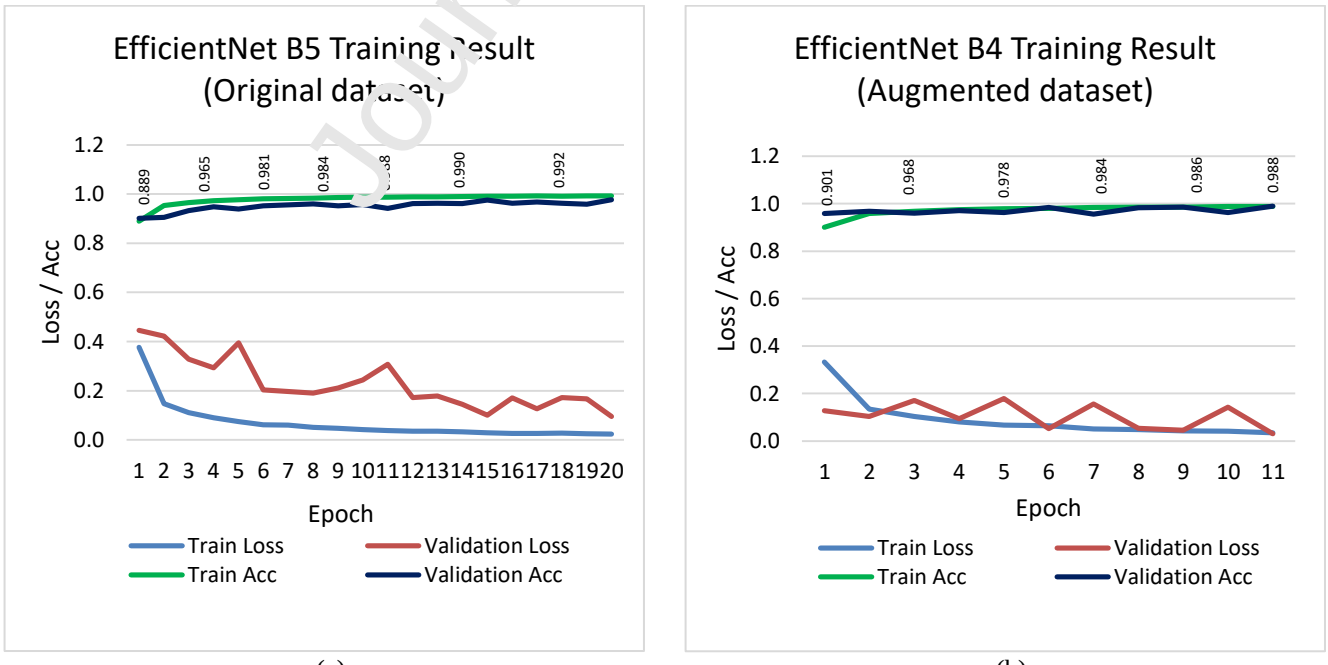


Figure 10. Accuracy/loss curves of the best models for training and validation sets of original (a) and augmented (b) datasets

The results obtained in the test dataset showed that B5 and B4 models of EfficientNet architecture have achieved the highest values compared to other deep learning models in original and augmented datasets with 99.91% and 99.97% respectively for accuracy and 98.42% and 99.39% respectively for precision. Although the accuracy values offered by the models varied between 99.45% and 99.97% in a narrow range, when it was evaluated in terms of the precision metric, it was seen that these values varied with more significant differences between 90.26% and 99.39%. Precision is the proportion of correctly predicted positives out of all samples predicted as positive by the model and it increases as the number of false positives decreases. While all models used in the study increased accuracy and precision with varying differences on the augmented dataset, the B4 model that offered the best performance on the augmented dataset achieved improvement in the accuracy and precision with 0.13% and 2.16% respectively. This showed us that as the number of data increased, the number of false positive predictions of models decreased and precision increased.

In order to more perceptibly demonstrate the performance of deep learning architectures discussed in this study, total incorrect classification numbers of the models for all classes and incorrect classification numbers for two notable classes were given in Table 8. As can be seen, B5 is the model with the lowest number of errors with 33 in the original dataset. For the augmented dataset, the model with the lowest number of incorrect classifications is B4 with 12. Two of the most remarkable number of misclassifications performed by the models were for healthy potato in the original dataset and for tomato early blight in the augmented dataset, respectively. As can be seen, while the B5 model made mistakes in only 9 samples in the classification of healthy potatoes, all other models made a relatively high number of misclassifications. In addition, B3, B4 and B7 models only have 1 misclassification in the augmented dataset, while other CNN architectures have a high number of misclassifications except VGG16.

Table 8. False prediction numbers of deep learning models

	B0	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**AlexNet**	**ResNet50**	**VGG16**	**Inception V3**
Number of false predictions for healthy potato in original dataset	37	43	37	27	42	**9**	41	28	**49**	**48**	**47**	**43**
Total false predictions for original dataset	73	78	64	63	62	**33**	77	54	208	89	73	73
Number of false predictions for tomato early blight in augmented dataset	3	2	3	1	**1**	7	3	1	**20**	**17**	**2**	**4**
Total false predictions for augmented dataset	45	30	34	23	**12**	25	26	39	124	47	24	27

There are many studies in the literature that classify plant disease using this dataset. In PlantVillage dataset, most of the studies (Amara et al., 2017; Brahimi et al., 2017; Cruz et al., 2017; Durmus et al., 2017; Rangarajan et al., 2018; Sibiya and Sumbwanyambe, 2019; Wallelign et al., 2018; Yamamoto et al., 2017; K. Zhang et al., 2018) classify the diseases of a single plant species. In addition, it was observed that some other studies not only used PlantVillage dataset but also some other plant images obtained from the real environment (Chen et al., 2020b; Ferentinos, 2018; KC et al., 2019; Khan et al., 2018; Ma et al., 2018; X. Zhang et al., 2018).

The results of studies (Chen et al., 2020b; Geetharamani and Pandian, 2019; Mohanty et al., 2016; Too et al., 2019) dealing with diseases of all plant species in PlantVillage dataset were compared with the results of our study and shown in Table 9. As seen in this table, EfficientNet B5 model achieved slightly better accuracy than studies of Mohanty et al., Chen et al. and Too et al. with original dataset. On the augmented dataset, EfficientNet B4 model was better than the model proposed by Geetharamani and Pandian in terms of accuracy and precision metrics. Experimental studies showed that EfficientNet B4 and B5 models, which offer the best classification performance, could be successfully used in plant leaf disease classification.

Table 9. Comparison of deep learning methods for plant leaf disease classification

Study	Method	Sen (%)	Spe (%)	Acc (%)	Pre (%)
Mohanty et al. (Mohanty et al., 2016)	GoogleNet (original dataset)	99.35	---	99.35	99.35
Too et al. (Too et al., 2019)	DenseNets-121 (original dataset)	---	---	99.75	---
Geetharamani and Pandian (Geetharamani and Pandian, 2019)	9-layer CNN model (augmented dataset)	99.89	---	96.46	96.47
Chen et al. (Chen et al., 2020b)	MobileNet-Beta (original dataset)	99.01	99.94	99.85	---
Our study	EfficientNet B5 (original dataset)	98.31	99.96	99.91	98.42
	EfficientNet B4 (augmented dataset)	99.38	99.98	99.97	99.39

5 Conclusions

Deep learning methods have recently become popular for image processing and pattern recognition. In this study, EfficientNet deep learning architecture was proposed to classify the plant leaf images of 39 classes in the PlantVillage dataset. The success of the proposed architecture was compared with the state-of-the-art deep learning architectures used in plant leaf disease detection in the literature. Experimental studies were conducted in both original and augmented versions of the PlantVillage dataset. Considering both the average accuracy and the average precision metric on both the original and augmented datasets, the B4 and B5 models were found to be superior to other CNN architectures. The B5 model achieved 99.91% accuracy and 98.42% precision in the original dataset, while the B4 model achieved 99.97% accuracy and 99.39% precision in the augmented dataset.

Furthermore, in EfficientNet architecture, even though the input image size was necessarily resized to 132x132 due to hardware limitations, it yielded more successful results than other CNN models that received input images with higher resolutions. On the other hand, when the training times of the models per epoch were analyzed, AlexNet had the lowest times in original and augmented datasets, with 310 and 352 seconds respectively, but it's average accuracy and precision were behind other models. In addition, B5 and B4 models, which achieved the highest accuracy and precision in the original and augmented datasets respectively, completed 1 epoch in acceptable times as 1930 and 1548 seconds respectively.

In the future work, it is planned to expand the plant leaf disease dataset by increasing the plant diversity and the number of classes. This will help developing models that can make more accurate predictions in difficult environments. By deploying these improved models in mobile environments, both plant pathologists and farmers will be able to quickly diagnose the plant diseases and take the necessary precautions.

References

Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., Alrahamneh, Z., 2011. Fast and accurate detection and classification of plant diseases. Int. J. Comput. Appl. 17, 31–38.

Amara, J., Bouaziz, B., Algergawy, A., 2017. A Deep Learning-based Approach for Banana Leaf Diseases Classification, in: Datenbanksysteme Für Business, Technologie Und Web (BTW 2017) - Workshopband. Bonn, pp. 79–88.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., Stefanovic, D., 2019. Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection. Symmetry (Basel). 11, 939. https://doi.org/10.3390/sym11070939

Brahimi, M., Boukhalfa, K., Moussaoui, A., 2017. Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. Appl. Artif. Intell. 31, 299–315. https://doi.org/10.1080/08839514.2017.1315516

Chen, J., Yin, H., Zhang, D., 2020a. A self-adaptive classification method for plant disease detection using GMDH-Logistic model. Sustain. Comput. Informatics Syst. 28, 100415. https://doi.org/10.1016/J.SUSCOM.2020.100415

Chen, J., Zhang, D., Nanehkaran, Y.A., 2020b. Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimed. Tools Appl. 1–19. https://doi.org/10.1007/s11042-020-09669-w

Chen, J., Zhang, D., Nanehkaran, Y.A., Li, D., 2020c. Detection of rice plant diseases based on deep transfer learning. J. Sci. Food Agric. 100, 3246–3256. https://doi.org/10.1002/jsfa.10365

Chen, Junde, Chen, Jinxiu, Zhang, D., Sun, Y., Nanehkaran, Y.A., 2020d. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 173, 105393. https://doi.org/10.1016/J.COMPAG.2020.105393

Cruz, A.C., Luvisi, A., De Bellis, L., Ampatzidis, Y., 2017. Vision-based plant disease detection system using transfer and deep learning, in: 2017 Asabe Annual International Meeting. p. 1.

Durmus, H., Gunes, E.O., Kirci, M., 2017. Disease detection on the leaves of the tomato plants by using deep learning, in: 2017 6th International Conference on Agro-Geoinformatics. IEEE, pp. 1–5. https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016

Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 145, 311–318. https://doi.org/10.1016/J.COMPAG.2018.01.009

Geetharamani, G., Pandian, A., 2019. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput. Electr. Eng. 76, 323–338. https://doi.org/10.1016/J.COMPELECENG.2019.04.011

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. 2015. arXiv Prepr. arXiv1512.03385.

Hughes, D.P., Salathe, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics.

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Yilong, Dong, Q., Shen, H., Wang, Yongjun, 2017. Artificial intelligence in healthcare: past, present and future. Stroke Vasc. Neurol. 2, 230–243. https://doi.org/10.1136/svn-2017-000101

Johannes, A., Picon, A., Alvarez-Gila, A., Echazarra, J., Rodriguez-Vaamonde, S., Navajas, A.D., Ortiz-Barredo, A., 2017. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput. Electron. Agric. 138, 200–209. https://doi.org/10.1016/j.compag.2017.04.013

KC, K., Yin, Z., Wu, M., Wu, Z., 2019. Depthwise separable convolution architectures for plant disease classification. Comput. Electron. Agric. 165, 104948. https://doi.org/10.1016/J.COMPAG.2019.104948

Khan, M.A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., Saba, T., 2018. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. Comput. Electron. Agric. 155, 220–236. https://doi.org/10.1016/J.COMPAG.2018.10.013

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput. 1, 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Li, M., Zhang, T., Chen, Y., Smola, A.J., 2014. Efficient mini-batch training for stochastic optimization, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '14. ACM Press, New York, New York, USA, pp. 661–670. https://doi.org/10.1145/2623330.2623612

Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., Sun, Z., 2018. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Comput. Electron. Agric. 154, 18–24. https://doi.org/10.1016/J.COMPAG.2018.08.048

Mark, S., Andrew, H., Menglong, Z., Andrey, Z., Liang-Chieh, C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1–10. https://doi.org/10.3389/fpls.2016.01419

Mokhtar, U., Ali, M.A.S., Hassanien, A.E., Hefny, H., 2015. Identifying Two of Tomatoes Leaf Viruses Using Support Vector Machine. Springer, New Delhi, pp. 771–782. https://doi.org/10.1007/978-81-322-2250-7_77

Nanehkaran, Y.A., Zhang, D., Chen, J., Tian, Y., Al-Nabhan, N., 2020. Recognition of plant leaf diseases based on computer vision. J. Ambient Intell. Humaniz. Comput. 1–18. https://doi.org/10.1007/s12652-020-02505-x

Pantazi, X.E., Moshou, D., Tamouridou, A.A., Kasderidis, S., 2016. Leaf Disease Recognition in Vine Plants Based on Local Binary Patterns and One Class Support Vector Machines. Springer, Cham, pp. 319–327. https://doi.org/10.1007/978-3-319-44944-9_27

Rangarajan, A.K., Purushothaman, R., Ramesh, A., 2018. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput. Sci. 133, 1040–1047. https://doi.org/10.1016/J.PROCS.2018.07.070

Revathi, P., Hemalatha, M., 2014. Identification of cotton diseases based on cross information gain deep forward neural network classifier with PSO feature selection. Int. J. Eng. Technol. 5, 4637–4642.

Rumpf, T., Mahlein, A.-K., Steiner, U., Oerke, E.-C., Dehne, H.-W., Plümer, L., 2010. Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. Comput. Electron. Agric. 74, 91–99. https://doi.org/10.1016/J.COMPAG.2010.06.009

Saleem, M.H., Potgieter, J., Mahmood Arif, K.M., 2019. Plant Disease Detection and Classification by Deep Learning. Plants 8, 468. https://doi.org/10.3390/plants8110468

Sankaran, S., Mishra, A., Ehsani, R., Davis, C., 2010. A review of advanced techniques for detecting plant diseases. Comput. Electron. Agric. 72, 1–13. https://doi.org/10.1016/J.COMPAG.2010.02.007

Shen, D., Wu, G., Suk, H.-I., 2017. Deep Learning in Medical Image Analysis. Annu. Rev. Biomed. Eng. 19, 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

Sibiya, M., Sumbwanyambe, M., 2019. A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. AgriEngineering 1, 119–131. https://doi.org/10.3390/agriengineering1010009

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv Prepr. arXiv1409.1556.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Comput. Intell. Neurosci. 2016. https://doi.org/10.1155/2016/3289801

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437. https://doi.org/10.1016/J.IPM.2009.03.002

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826.

Tan, M., Le, Q. V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. 36th Int. Conf. Mach. Learn. ICML 2019 2019-June, 10691–10700.

Too, E.C., Yujian, L., Njuki, S., Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. 161, 272–279. https://doi.org/10.1016/J.COMPAG.2018.03.032

Wallelign, S., Polceanu, M., Buche, C., 2018. Soybean Plant Disease Identification Using Convolutional Neural Network. Thirty-First Int. Flairs Conf.

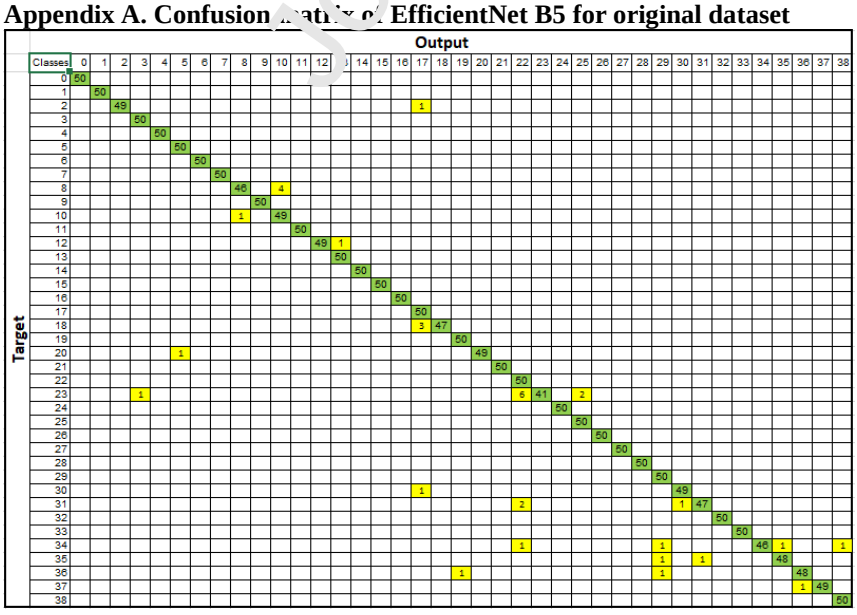
Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. J. Big Data 3, 9. https://doi.org/10.1186/s40537-016-0043-6

Yamamoto, K., Togami, T., Yamaguchi, N., 2017. Super-Resolution of Plant Disease Images for the Acceleration of Image-based Phenotyping and Vigor Diagnosis in Agriculture. Sensors 17, 2557. https://doi.org/10.3390/s17112557

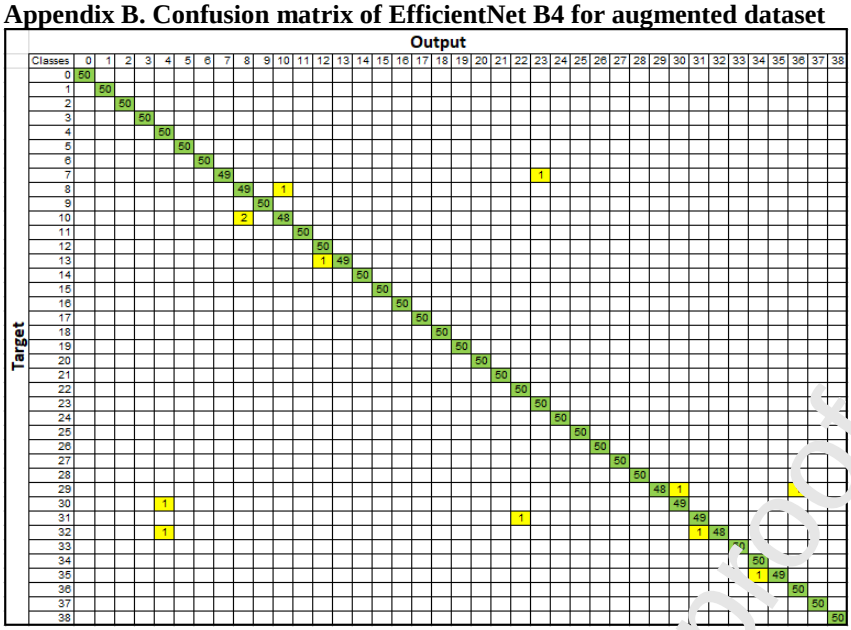
Zhang, K., Wu, Q., Liu, A., Meng, X., 2018. Can Deep Learning Identify Tomato Leaf Disease? Adv. Multimed. 2018, 1–10. https://doi.org/10.1155/2018/6710865

Zhang, X., Qiao, Y., Meng, F., Fan, C., Zhang, M., 2018. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. IEEE Access 6, 30370–30377. https://doi.org/10.1109/ACCESS.2018.2844405

Appendix A. Confusion matrix of EfficientNet B5 for original dataset



Appendix B. Confusion matrix of EfficientNet B4 for augmented dataset



Author Declaration

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Highlights:

- EfficientNet architecture was proposed for plant leaf disease classification.
- The PlantVillage dataset containing 55448 images with 39 classes was trained.
- Proposed model was compared with other state-of-the-art deep learning models.
- All models were trained using transfer learning technique.
- EfficientNet B5 and B4 models were superior to other models in terms of accuracy.