

AFD-Net: Apple Foliar Disease Multi Classification using Deep Learning on Plant Pathology Dataset

Anju Yadav

Udit Thakur

Rahul Saxena

Vipin Pal

Vikrant Bhateja

Jerry Chun-Wei Lin (✉ jerrylin@ieee.org)

Western Norway University of Applied Sciences <https://orcid.org/0000-0001-8768-9709>

Research Article

Keywords: Computer vision, Deep learning, Kaggle, Foliar disease, Efficient net.

Posted Date: January 13th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1158879/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

AFD-Net: Apple Foliar Disease Multi Classification using Deep Learning on Plant Pathology Dataset

Anju Yadav¹, Udit Thakur¹, Rahul Saxena¹, Vipin Pal², Vikrant Bhateja³ and Jerry Chun-Wei Lin⁴

¹Manipal University Jaipur, India

² National Institute of Technology, Meghalaya, India

³Shri Ramswaroop Memorial Group of Professional Colleges, India

³Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India

⁴Western Norway University of Applied Sciences, Bergen, Norway

anju.yadav@jaipur.manipal.edu, udit.thaku@jaipur.manipal.edu,
rahul.saxena@jaipur.manipal.edu, vipirwr@gmail.com, bhateja.vikrant@ieee.org,
jerrylin@ieee.org

Abstract. Plant diseases significantly affect the crop, so their identification is very important. Correct identification of these diseases is crucial for establishing a good disease control strategy to avoid time and financial losses. In general, machines can greatly reduce the possibility of human error. In particular, computer vision techniques developed through deep learning have paved a way to detect and diagnose these plant diseases on the leaf. In this work, the model AFD-Net was developed to detect and identify various leaf diseases in apple trees. The dataset is from Kaggle 2020 and 2021 and was financially supported by the Cornell Initiative for Digital Agriculture. A AFD-Net was proposed for leaf disease classification in apple trees and the results of the efficiency of the model are compared with other state-of-the-art deep learning approaches. The results of the experiments in the validation dataset show that the proposed AFD-Net model achieves the highest values compared to other deep learning models in the original and extended datasets with 98.7% accuracy for Plant Pathology 2020 and 92.6% for Plant Pathology 2021.

Keywords: Computer vision, Deep learning, Kaggle, Foliar disease, Efficient net.

1 Introduction

Leaf diseases of apples refer to the common diseases found on apple leaves, namely scab, rust, and powdery mildew, among others [1]. Apple scab, caused by a fungal pathogen, is one of the most economically important fungal diseases of apple in the world [2,3]. The symptoms of apple scab are clearly visible fungal structures on the surface of the leaf. Rust disease also causes severe losses when

environmental conditions are favorable for disease growth. For example, in a plant affected by Rust [4], small yellow spots appear on the leaf surface.

Farmers spend a lot of money on disease control and inadequate technical support, but the results always lead to poor disease control. Foliar diseases spread rapidly and can destroy large portions of the yield in a very short time. In some cases, these diseases destroy the entire crop if the disease is not controlled quickly and accurately. Foliar diseases are a challenge to crop production in most countries. They reduce crop yields, fruit quality, and nutritional value, resulting in lower returns for the farmers [4].

Machine learning models learn, recognize patterns, and make decisions with minimal human intervention. Ideally, machines increase accuracy and efficiency and eliminate the possibility of human error [5]. The use of AI in agriculture helps farmers gain insights into their crops and use the data to increase their overall production. Various computer vision techniques can be applied to gain the desired insights [6,7].

Recent advances in computer vision enabled by deep learning have paved the way for more accurate disease diagnosis. Using large public datasets of diseased and healthy plants and leaves, a CNN can be trained to identify various leaf diseases [8,2]. With the increasing availability of smartphones, the approach of training deep-learning models on a large scale has emerged as a clear way to diagnose crops on a large scale [5]. Every year, farmers worldwide are affected by foliar diseases. Our research could contribute greatly to the automation of disease detection worldwide and potentially help millions of people [1].

Today's major crops are plagued by a variety of diseases. Diseases in crops can occur in several areas of the plant, including the roots, leaves, and stem, although the leaves are the most typical site for disease detection. It is difficult to detect and diagnose diseases because leaves have a variety of sizes, shapes, and colors. The article deals with various aspects of diseases and classifies them based on the characteristics of the condition of the leaves. It is important to identify the cause at the root, which is beneficial and time-saving for both the agricultural sector as well as the farmers. So far, to our knowledge, there are some works on leaf disease detection in apples, and these datasets have fewer classes[2,5]. Only one work has looked at the 2020 plant pathology dataset, but it has the limitation of lacking augmentation techniques. Also, there was no cross-validation to determine the out-of-fold predictions and the scores of the respective folds [1]. This could lead to an overfitting problem in some cases because the model does not have adequate validation for evaluation.

Our approach is to first classify a given image from a test dataset to identify the conditions of the plants (i.e., diseased or healthy). Moreover, the dataset is preprocessed and extensions are made. After that, different diseases can be identified and sorted out using the proposed model. The results obtained by applying the proposed model can lead to multiple diseases on a plant (i.e., more than one). The main contribution of this paper is as follows:

- This article discusses the state-of-the-art of machine learning and deep learning applications in disease identification and classification.

- A novel AFD-Net model is proposed to automate the detection and multiple classification of Apple foliar diseases using Hybrid EfficientNet Model.
- We collected and preprocessed the apple leaf disease dataset, i.e. Plant Pathology 2020 and 2021. Moreover, the proposed model achieves promising results compared to the currently available methods for apple leaf disease detection and classification.

In addition, section 2 provides a detailed literature review on leaf disease detection and classification in apples using machine learning and deep learning methods. In section 3, the proposed methodology “AFD-Net” and the evaluation parameters are explained in detail. Moreover, in section 4, the description of the dataset and its preprocessing steps are discussed. In section 5, the design of the implementation and the analysis of the results are described and discussed. Finally, conclusions are drawn and future work is described in section 6.

2 Literature Review

In recent years, ML and DL are widely used to detect plant diseases, which helps farmers identify the right foliar disease and apply the appropriate treatments [9]. Digital images are widely used in computer vision to identify the diseases for further classification based on their symptoms [10,11,12,13]. However, it is challenging to accurately identify disease from leaves due in part to the resolution, background light, and shadows of the leaves, among other[14]. Machine learning (ML) and deep learning (DL) approaches are well suited for processing image data, especially in agriculture, and can be used to detect and classify plant diseases from the collected images (i.e., photos of leaves) [15].

Agarwal et al. proposed a model consisting of 3 maximal pooling layers followed by two densely connected layers. After testing with different numbers of convolutional layers from 2 to 6, it was found that 3 layers provide the best accuracy [2]. The proposed model achieves a very impressive accuracy, i.e., 96%. The database used for the developed framework consists of nearly 50,000 images of 171 diseases, including 21 plant species. The original samples were divided into smaller images containing individual lesions or localized symptom regions. This was done to increase the size of the dataset and to test how the CNN would perform with more localized information.

Instead of taking pictures in the natural condition, Zhong and Zhao took pictures with a solid background [5]. Images of all symptoms were resized to 128 by 128. The dataset was split 8:2 for the training and test datasets by randomly selecting images from the dataset. After deduplication, the dataset contained 2,462 images, with 85% of the images used for training and 15% for validation. The accuracy of this method for the test dataset was 93.71%.

Militante et al. proposed a model, i.e., a combination of a convolutional layer, an activation layer, a pooling layer, and a fully connected layer [6]. The images used in this study were in color and were reduced to 96 by 96 for further processing. An accuracy of 96.5% was achieved with 75 epochs while the model

was well trained. A maximum accuracy of 100% was achieved when random images of plant varieties and diseases were tested.

Arsenovic et al. presented a dataset of 79,265 images. Traditional augmentation methods and generative adversarial networks are used for image augmentation. Moreover, a 2-stage NN architecture was proposed for classification and a test accuracy of 93.67% was achieved with the trained model. The DCNN model [16] was presented for detecting the leaf disease of apple tree by combining DenseNet and Xception. The results show that the developed model achieved 98% accuracy.

Yu et al. designed two subnetworks; the first is used for segmentation to identify features, and the second model is used for classification. In the experiments, the proposed model provided an accuracy of 89.4% [17]. The use of AI in agriculture helps farmers gain insights about their crops and use this data to increase their production wisely. Using the proposed methods, Thapa et al. captured 3,650 of high-resolution images of several apple leaf diseases and annotated the dataset with the help of an expert in the field of pathology to confirm the annotations for the images, which were difficult to distinguish based on symptoms[1]. The overall test accuracy achieved by a ResNet50 network pre-trained on ImageNet was 97%.

Raschka et al. proposed a CNN model and achieved 97.62% accuracy in identifying four different types of apple leaf blight, detecting infected parts on the leaf, and classifying between healthy and infected fruits, e.g., apples [7]. However, there are very few studies dealing with apple foliar disease and most of them are limited to a specific type of disease, either biotic or abiotic.

3 Proposed Methodology

In this section, a framework called Apple Foliar Disease Neural Network (AFD-Net) is presented in this paper. Basically, our work is based on the transfer learning approach, where the first and most important step is to collect the dataset [1]. The second step is to project and clean the database using image processing steps [18] to find outliers and class imbalances. This was followed by an exploratory data analysis of the dataset with all graphs and class distribution of the foliar diseases. Extensions such as rotation, transformation, and flips were applied to increase the diversity/learning capability of the model [19,20]. After preprocessing all the data, the data were fed into the training pipeline with 5-fold cross-validation, properly validating the training data. These iterations are repeated continuously until we find a stable cross-validation value for our training data that matches the test data.

To improve the performance of the model, several strategies can be considered, e.g., hyper-parameter tuning approaches, changing the loss function such as bi-tempered loss for noisy labels, changing the learning rate, and freezing/unfreezing the model layers. Different ensemble approaches (weighted average, normal average, rank aggregate) were also considered to increase the prediction accuracy. In the designed model, the weighted average strategy was chosen as the model

for the designed approach. The flowchart for the above methodology is shown in Fig. 1.

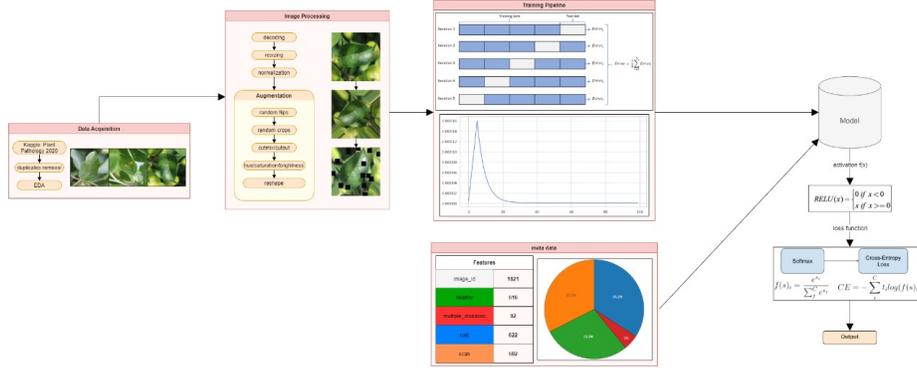


Fig. 1: Flowchart of proposed methodology

3.1 Efficient Net distribution

The performance of the models used in the ImageNet dataset has increased since 2012 [21] as they have become more complex, but many of them are not effective in terms of computational load. The EfficientNet model [21], which is one of the best models since it achieves 84.4% accuracy on the ImageNet classification problem with 66 M parameters, can be considered as a group of CNN models. It consists of 8 models between B0 and B7, and as the number of models increases, the number of computed parameters does not increase significantly while the accuracy increases noticeably[21]. We used only B3 and B4 to find a middle ground, as they performed the best under various experiments and hardware constraints. Unlike other CNN models, EfficientNet uses a new activation function called Swish instead of the Rectifier Linear Unit (ReLU) activation function [21]. From the experiments, simply substituting ReLU units with Swish units improves the classification accuracy in ImageNet by 0.6% for Inception-ResNet-V2; it outperforms ReLU in many deep neural networks, as shown in Fig. 2.

3.2 Architecture of the proposed AFD-Net

In the proposed AFD-Net model, EfficientNet B3 and B4 are attached along with the lambda layer, which is capable of running on few parameters to obtain great results with ensemble of the probabilities from both the architectures. At the end Dropout layers were added to the above functional layers of the model, and ended with softmax output function. In total, there were 28.2M trainable parameters in the experiments. The AFD-Net achieve both high accuracy and

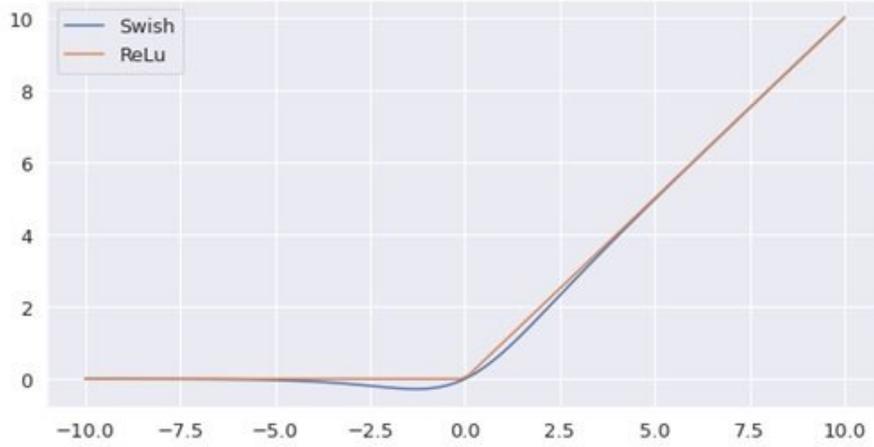


Fig. 2: Swish versus ReLU $F(x)$

efficiency over pre-existing CNN based models, this reducing parameter size and achieving the accuracy. The architecture of developed *AFD-Net* can be seen in Fig. 3.

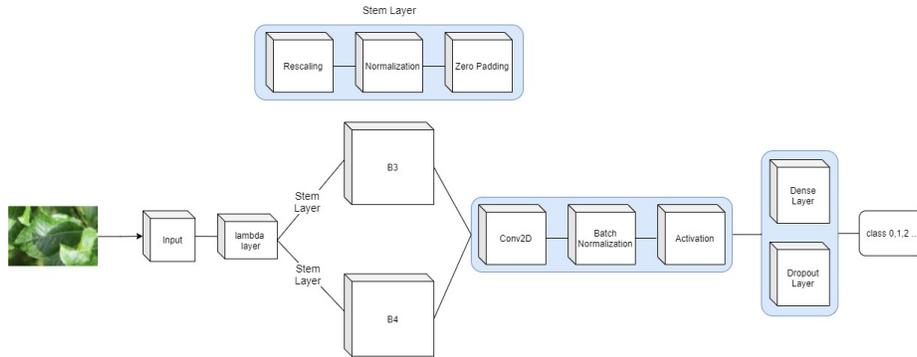


Fig. 3: Proposed AFD-Net architecture

In this proposed model of lambda layers, we connected two functional layers of efficient nets B3 and B4, followed by Dropout layer to prevent overfitting of the model, and at the end we added a Dense layer of 4 units together with a **softmax** activation function for the prediction part. The developed AFD-Net is then described in Algorithm 1.

The algorithm takes as input the set of images as t-f records with labels. These images are passed through the augmentation module to extract a variety of augmented images that are fed into the training module of the *AFD-Net*. The

Algorithm 1 Proposed AFD-Net

Input: IMG is the set of images as the training data set and fed through tf-records with labels; N is the number of epochs; K is the number of folds, and oof is out of fold predictions.

Output: the trained model, m .

- 1: $X \leftarrow IMG$;
- 2: $Aug_X \leftarrow Aug_Process(X)$; {generate augmented images}
- 3: $Y \leftarrow Cov_Vec(Aug_X)$; {convert images into one-hot vectors of lables}
- 4: $\{(trainX, trainY), (valX, valY)\} \leftarrow Kfold(X)$; {n_splits:5, random:42}
- 5: **for** each epoch till N **do**
- 6: $lr_scheduler \leftarrow LR(callback)$; {setting up learning rate}
- 7: $model \leftarrow AFDNet(B3, B4, lambda, adam, Aug_X, label_smooth)$; {building model}
- 8: $oof \leftarrow Predict(model, val_dataset)$; {cross validation predictions}
- 9: $dataframe \leftarrow AppendOofs(y_true, y_pred)$; {creating data frame of predictions}
- 10: **end for**
- 11: $m \leftarrow save((model, dataframe)|t = 1, 2, \dots, K)$ {saving the model}

augmentations are of different types and range from flipping to cropping. The resulting augmented images and their associated labels are then converted into one-hot encoding vectors with labels to incorporate the categorical cross-entropy loss (lines 1 to 2). After these preprocessing steps, the data are split into a training set and a validation/test set using K-fold cross-validation with 5 folds stratified by labels (line 4). In lines 5 to 10, each tuple of the training dataset is fed into the *AFD-Net* model, setting the learning rate and other hyperparameters of the model (line 6). In the *AFD-Net* model, EfficientNet B3 and B4 are attached along with the lambda layer, which can work with few parameters to produce great results with an ensemble of probabilities from both architectures (line 7). In the end, Dropout layers were added to the above functional layers of the model and finished with the `softmax` output function to generate probabilities for each label/class. Finally, the validation/test tuples are used to make the predictions about the model and store the results in a data frame (lines 8 to 9). Finally, the trained model is saved to disk after all epochs are completed (line 11).

3.3 Evaluation metrics

The efficiency of the proposed model is evaluated with 5-fold cross-validation using stratified splits [22]. The dataset was split into two parts, i.e., a training set and a validation set. The validation part is used to test the model, as can be seen in Fig. 4.

The performance metrics selected in this study are commonly used to measure model efficiency and performance, including accuracy, confusion matrix, specificity, sensitivity, precision, F1-score, and Matthews correlation coefficient (MCC). The determined parameters of these metrics are based on the rates of true positive (x), true negative (y), false positive (z), and false negative (w), as shown in Table 1.



Fig. 4: 5-Fold split cross validation

Table 1: The metrics of evaluation performance

Metric	Formula
Sensitivity	$x/(x + w)$
Specificity	$y/(y + z)$
Precision	$x/(x + z)$
F1-Score	$2x/(2x + z + w)$
Accuracy	$(x + y)/(x + w + z + y)$
MCC	$(x \times y - z \times w) / ((x + z)(x + w)(y + z)(y + w))^{0.5}$

3.4 Transfer learning algorithm for comparison

Inception-v3: Inception-v3 [23] is a design of a convolutional neural network from the Inception family that includes label smoothing, factorized 7×7 convolutions, and the inclusion of an auxiliary classifier to propagate label information further down the network. Inception-v3 proposed a method for regularizing the classification layer by computing the marginalized impact of label dropout during training. The smoothing of the label prevents overfitting of the model.

$$newTargets = (1 - \epsilon) * one - hot - targets + \epsilon/N, \quad (1)$$

where ϵ is a hyper parameter and set as 0.1 and N is the number of classes and set it as 4.

ResNet50 and ResNet101: When training deep networks, there comes a moment when accuracy reaches a saturation point and then rapidly degrades. This shows that not all neural network designs can be optimized equally well. To solve this problem, ResNet [24] employs a method known as “residual mapping”. The residual network allows these layers to explicitly match a residual mapping, rather than trusting that all pairs of stacked layers would match a desired underlying mapping. The building block of a residual network is shown below in Fig. 5.

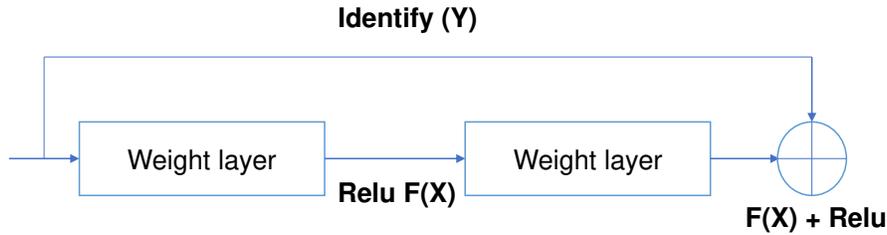


Fig. 5: ResNet building block

In addition, the feedforward neural networks with shortcut connections can be then formalized as:

$$F(y) + y \quad (2)$$

VGG-16: Convolutional layers one and two have a kernel of 64 features with 3×3 filter size. Convolutional layers three and four have a kernel of 124 features with 3×3 filter size. Two layers are followed by a 2 max-pool layer, so $56 \times 56 \times 128$ will be the reduced output of the results. For the fifth, sixth and seventh layers, 256 feature maps with a size of 3×3 are used. Followed by 2 max-pool layers, 512 filters will be used for the eighth through thirteen convolutional layers with kernel size 3×3 . Followed by one max-pool layer. There are fully connected hidden layers with 4096 units, followed by a softmax output layer with 1,000 units [24].

4 Dataset Description and Pre-processing

The dataset [1] used in this paper was taken from the Plant Pathology 2020 - FGVC7 Kaggle competition, which was financially supported by the Cornell Initiative for Digital Agriculture (CIDA). The dataset consists of 1,821 portrait and landscape images. The image size is either $2,048 \times 1,368$ pixels or $1,368 \times 2,048$. The second dataset is that of the Plant Pathology 2021-FGVC8 competition with a pilot dataset of 3,651 RGB images of leaf diseases in apples. For Plant Pathology 2021-FGVC8¹, the number of leaf disease images was significantly increased and additional disease categories were added. The dataset contains 18,633 of high-quality RGB images of leaf diseases in apples, including a large, expert-annotated disease dataset. This dataset reflects real field scenarios by representing inhomogeneous backgrounds of leaf images taken at different stages of ripening and at different times of day with different camera settings. We generated images of dimensions 512×512 with two string and `int` byte structures for images, names, and targets. The generated images were set to 100% quality. Along with the images, the partitioning was layered into 15 folds to maintain the proper class balance throughout the dataset.

4.1 Target distribution



Fig. 6: Target distribution of dataset

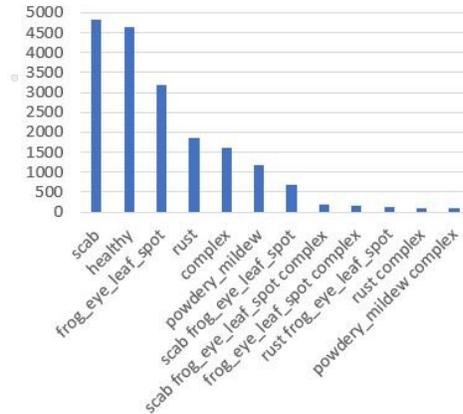


Fig. 7: Plant Pathology 2021

We have 4 categories of foliage in the Plant Pathology 2020 data set, including “healthy”, “rust”, “scab”, and “multiple Diseases”. “rust” is the prominent disease, followed by “scab”, “healthy”, and the lowest number of “multiple Diseases”. It was noted that the data were unbalanced with respect to the Multiple Diseases

¹ <https://www.kaggle.com/c/plant-pathology-2021-fgvc8>

class (see Fig. 6). In addition, the 2021 plant pathology dataset is categorized into 12 target classes. All target classes and their number distributions are shown in Fig. 7.

4.2 Channel distribution findings

Green is the most prominent color in the leaf record, which makes sense since the leaves are colored green. There was a lot of variance in the data set for the red and blue channels in both cases. The variance occurred in the infected leaves that were affected by scab, rust, or multiple diseases. The channel values appear to have an approximately normal distribution centered around 105. The green portions of the image have very low blue values, but in contrast, the brown portions have high blue values. This indicates that the green (healthy) parts of the image have low blue values, while the unhealthy parts tend to have high blue values. An unhealthy leaf with its RGB values is shown in Fig. 8.



Fig. 8: Unhealthy leaf with its RGB values of the disease

4.3 Data Splitting

In this study, the original datasets of apple leaf diseases “Plant Pathology 2020” and “Plant Pathology 2021” are used. The datasets are randomly divided into training and validation sets, i.e., 80% and 20%, respectively. The training and validation sets were used only for fitting the model and calculating the validation metrics, which can be seen in Table 2.

The original apple foliar disease dataset “Plant Pathology 2020” and “Plant Pathology 2021” are used in this study. The dataset are randomly divided

into training and validation sets i.e., 80% and 20%, respectively. Training and validation sets were only used for fitting of the model and calculating validation metrics respectively see Table 2.

Table 2: Dataset splitting

Data	Total	Training (80%)	Validation (20%)
Plant Pathology 2020	1,821	1,457	364
Plant Pathology 2021	18,633	14,905	3,727

For cross-validation, we used 5 folds split across 15 training datasets, each consisting of 120 targets (approx.) - healthy, rust, scab, and multiple diseases, respectively, for the original dataset. The training and validation datasets were only used to fit the model and calculate out-of-fold predictions, respectively.

4.4 Cutmix, Mixup and Basic Augmentations

To help the model learn all the outliers and reveal the state of the leaves, it must be trained with different augmentation images to increase its robustness. This will make the model more flexible to the newly injected and eccentric data. Different types of augmentations can be used for different problems. For the designed model, we used the basic augmentations provided by the Keras module - random (crop, hue, saturation, brightness and contrast) and shear transform, as shown in Fig. 9.

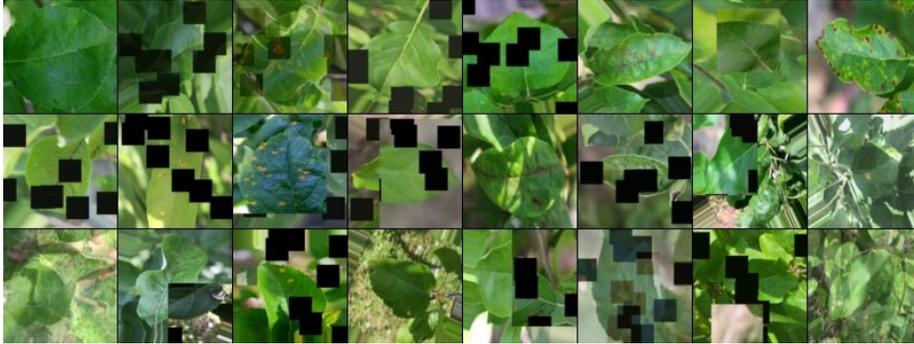


Fig. 9: Cutmix, Mixup and Basic Augmentations

The cutMix [19] and mixup [20] are the two augmentation techniques we implemented in the dataset. In the cutmix algorithm, part of the image is appended/attached to the other image to improve the localization of the model. Instead of simply cutting out pixels, as is the case with cutout or dropout, we

replace the cutout regions with a patch from another image. The ground truth labels are blended in proportion to the pixel count of the composite image. By asking the model to identify the object from a partial view, the additional patches can improve localization. In mixup expansion, two samples are mixed together by linear interpolation of their images and labels. Mixup samples suffer from unrealistic output and label ambiguity, and thus cannot perform well in tasks such as image localization and object recognition. Mixup alleviates this problem by mixing different features together, preventing a network from having too much confidence in the association between features and labels. Dropout/Cutout augmentation is a type of regional dropout strategy in which a random patch from an image is zeroed out (replaced with black pixels). Cutout samples suffer from reduction in information and regularization ability.

5 Experimental Evaluation

In this section, we discuss the detailed implementation of the proposed AFD-Net model. It was implemented using Google Colab and trained with TPUs. The model was tested with several k-folds, of which the 5-folds are the most efficient in terms of metrics. The custom learning scheduler was also optimized based on ramp-up epochs and decay value. Initially, we started with the categorical cross-entropy as the loss function with a label smoothing of 0.05 and then switched to the bi-tempered logistic loss to compare the model performance.

5.1 Network training

Before training the network on the leaf dataset, we used pre-trained weights from ImageNet [21] and noisy-student. The dataset with image size 512×512 was obtained from the training pipeline created for the purpose of pre-fetching tf-records with features such as caching and image decoding. The network was trained with a custom learning rate scheduler applied to the optimizer “adam”. We used two loss functions such as ‘categorical cross-entropy loss’ and ‘bi-tempered logistic loss’ as benchmarks. For the cv scheme, the model was trained on 80 – 20 splits of the training data for 40 epochs and then trained on 5-fold cv with 15-stratified splits across classes for 100 epochs per fold. During training, the label smoothing parameter in the loss functions helped the model stabilize and prune the predictions.

5.2 Performance comparison

Table 3 shows the various models tried with the image data. Comparing the different models from Table 3, it can be seen that using cutmixup can lead to a significant drop in accuracy compared to dropout augmentation, i.e. 88.4% from 96.7%. In addition, using efficient nets B3 and B4% separately leads to lower accuracy compared to combining these two i.e. 91.8% and 93.2%, respectively. It can also be inferred that the weights of noisy-student were better at extracting

features of the dataset and provided a good accuracy of 98.7%, as can be seen in Table 3. The parameters were tuned for the training phase of the proposed models and their variations. The stack size and seed were kept the same for all experiments to obtain consistent results. The accuracy results given below for the tuned models are considered as the average over all 5 folds.

Table 3: Comparative analysis of varied AFD-Net models

Model	Validation Accuracy	K-fold	Validation (20%)
AFD-Net-ImageNet	98.3 %	-	364
AFD-Net-Noisy Student	98.7 %	-	3,727
AFD-Net CutMixUp-ImageNet	88.4 %	-	
AFD-Net Dropout-ImageNet	96.7 %	-	
EfficientNet B3-ImageNet	91.8 %	5	
EfficientNet B4-ImageNet	93.2 %	5	
AFD-Net-ImageNet	95.8 %	5	
AFD-Net-Noisy Student	96.3 %	5	

The algorithm stated takes as input the set of images as t-f records with labels. The images ar

Although existing deep learning networks such as VGG, Inception, and ResNet can be used for leaf disease classification, these models have limitations in improving discriminatory power because they do not account for the mechanism of spatial attention to extract discriminating features between diseased and non-diseased areas. The accuracy of the AFD-Net model (98.7%) outperforms the existing TL models: ResNet 50(95.2%), ResNet (101 96.3%), VGG16 (96.7%), and Inception (95.6%). Therefore, the proposed “**AFD-Net-Noisy Student**” outperforms other state-of-the-art transfer learning approaches, which can be seen in Table 4.

Table 4: Comparative analysis of proposed model AFD-Net with other transfer learning models in the Plant Pathology 2020 dataset

Transfer Learning Models	Validation Accuracy
ResNet 50	95.2 %
ResNet 101	96.3 %
VGG16	96.7 %
Inceptionv3	95.6 %
Proposed AFD-Net	98.7%

In addition, the “Plant Pathology 2021” dataset is used to analyze the performance of the AFD-Net model. From Table 5, the accuracy of the proposed model is 92.6%, outperforming the other TL models and the three winners of the competition, i.e., for 1st place 88.3%, 2nd place 87.98%, and 3rd place 87.56%.

Table 5: Comparative analysis of proposed AFD-Net with other TL models in the Plant Pathology 2021 dataset

Transfer Learning Model	Validation Accuracy
Proposed AFD-Net	92.6%
ResNet 50	85.2 %
EfficientNet B3 - ImageNet	92.1 %
EfficientNet B4 - ImageNet	91.4 %
Inceptionv3	80.3 %
Kaggle (1st place)	88.3%
Kaggle (2nd place)	87.98%
Kaggle (3rd place)	87.56%

5.3 Comparison of proposed AFD-Net with existing approaches

In the literature review, we saw that some authors have worked on apple foliar disease detection. Most authors have worked with different data sets and some with small data sets. To our knowledge, only one article has worked with a plant pathology dataset and achieved an accuracy of 97%. Also, we compared our proposed model with existing models [1,2,5,17] for apple foliar disease detection and it is observed that presented AFD-Net achieved an accuracy of 98.7%, which can be seen in Table 6.

Table 6: Comparative analysis with different approaches

Model	Dataset	Accuracy
Proposed AFD-Net	Plant Pathology 2020	98.7%
Ranjita Thapa et. al. [4]	Plant Pathology 2020	97%
M. Agarwal et. al. [5]	Plant Village	96%
Zhong et. al. [7]	AIChALLENGER-Plant-Disease-Recognition	93.71%
Hee-Jin et. al. [9]	Apple Research Institute located in Republic of Korea	89.4%

5.4 Inferential statistical analysis

Inferential statistics were performed on “AFD-Net: noisy-student (cross-validation)”. To evaluate the performance of the designed model, we decided to take the out-of-fold predictions from each fold and compute the previously mentioned metrics. Training of the model was started with a custom learning rate scheduler to maximize validation accuracy and minimize loss as the model converged. The model was consistent after the warm-up epochs and did not deviate significantly from detecting underfitting or overfitting. Although there was a minority in the fourth class, i.e., “multiple diseases”, the model yielded an accuracy of 0.78. From the confusion matrix, it is clear that the model had difficulty with the “multiple diseases” class due to the imbalance between the classes, which is why the recall

value for this class was 0.55, as shown in Fig.10. The curve for accuracy versus recall is also shown in Fig. 10.

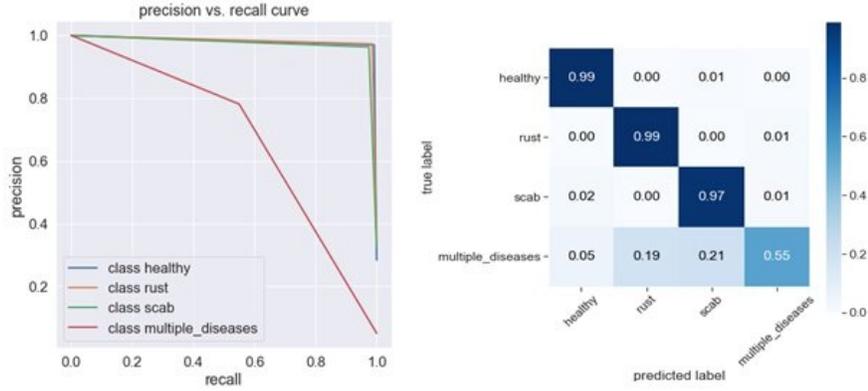


Fig. 10: Precision vs recall curve and Confusion matrix

Figs. 11 and 12 show the training results of the model using 5-fold splits. Of the 5 folds, the best fold was considered. For further analysis, we plotted the ROC curve for our proposed AFD-Net for different classes and it was found that the accuracy is around 99% for all classes except for multiple classes as shown in Fig. 13.

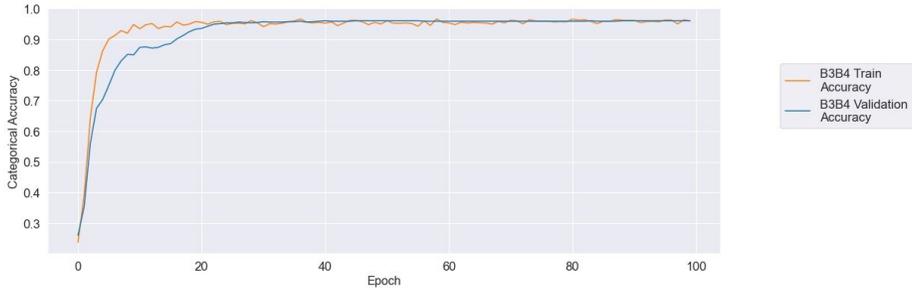


Fig. 11: Best fold training accuracy curve

In addition, we have calculated a quantitative analysis of the parameters of the proposed model. From the Table 7, we can infer the evaluation metrics. Sensitivity is another name for Recall. It is a measure of the proportion of actual positive cases that were predicted to be positive. In our case, the sensitivity is 1, which means that the proportion of true positives is higher than the proportion of false negatives. Similarly, for specificity, a higher value leads to a higher

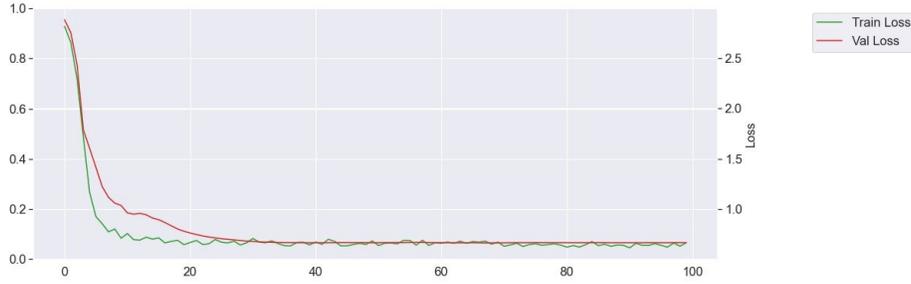


Fig. 12: Best fold training loss curve

proportion of true negative cases and a lower rate of false positives. For the measure of accuracy, the values for each class with the lowest score are unique for the “multiple diseases” classification. For the proposed AFD-Net model with LR $5e-06$ obtains the precision and recall value close to 1, except for multiple diseases and also the F1 score compared to other loss functions.

Table 7: Classification results of the proposed model with different loss functions

Different loss function		Precision	Recall	F1-Score	Support
AFD-Net with LR $5e-06$	Healthy	0.97	0.99	0.98	516
	Rust	0.97	0.99	0.98	622
	Scab	0.96	0.97	0.97	592
	Multiple diseases	0.78	0.55	0.65	91
AFD-Net with Bi-tempered loss	Healthy	0.97	0.99	0.98	516
	Rust	0.97	0.99	0.98	622
	Scab	0.96	0.98	0.97	592
	Multiple diseases	0.80	0.56	0.66	91
AFD-Net with LR $5e-03$	Healthy	0.97	0.99	0.98	516
	Rust	0.96	0.98	0.97	622
	Scab	0.95	0.97	0.96	592
	Multiple diseases	0.79	0.53	0.63	91

Besides, the performance of all the models is compared with that of the proposed model AFD-Net and from the results of the Table 8, it can be seen that the proposed model performs better than the other approaches. For our proposed model, the values of sensitivity, specificity, precision, F1 score, accuracy and MCC are 0.99, 0.97, 0.89, 98.7 and 0.94, respectively. These values proved to be better than those of other models. For better illustration, we have also plotted a graph for all the models. It can be clearly seen that the performance of our proposed model is better than the other models for both loss functions, i.e., LR $5e-06$ and LR $5e-03$, that can be observed from Figs. 14 and 15.

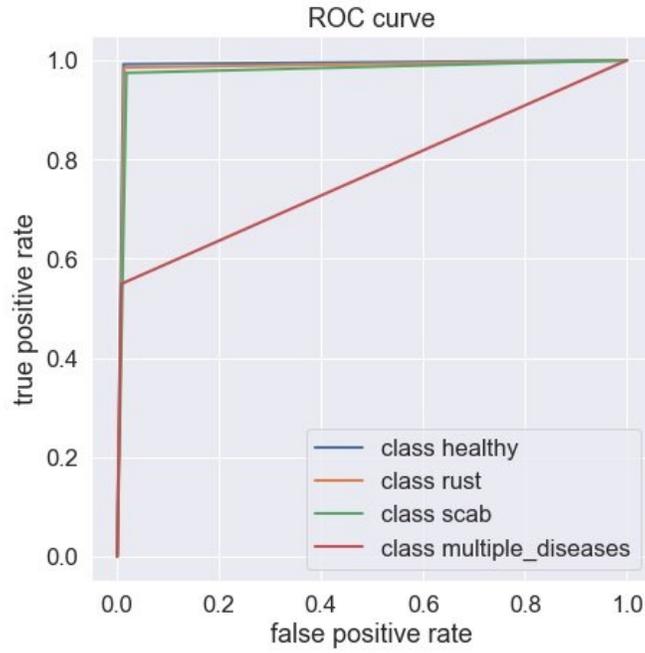


Fig. 13: ROC curve of proposed AFD-Net

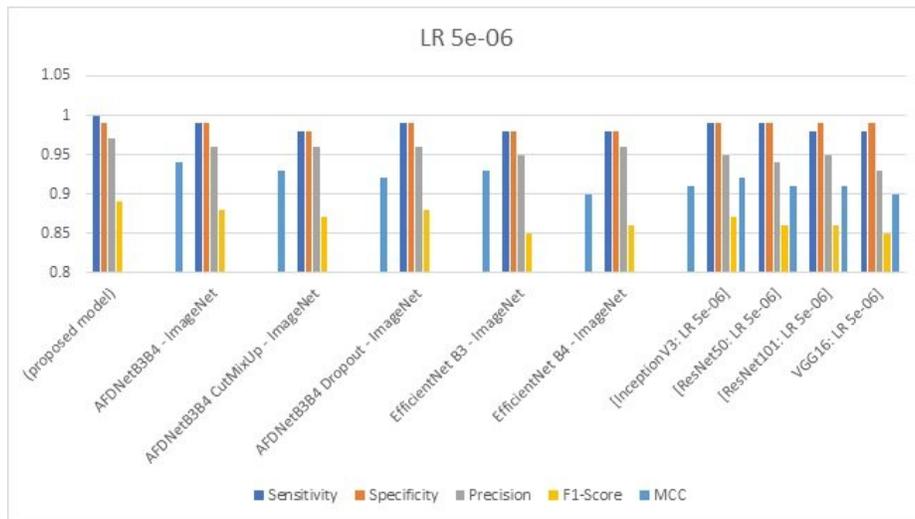


Fig. 14: Comparison of proposed “AFD-Net-Noisy Student” with other TL models for LR 5e-06

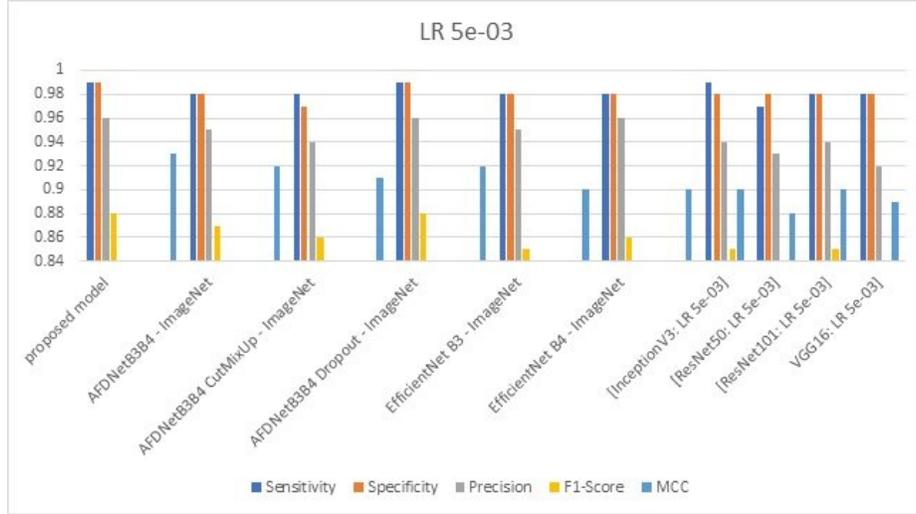


Fig. 15: Comparison of proposed “AFD-Net-Noisy Student” with other TL models for LR 5e-03

Table 8: Comparison of proposed “AFD-Net-Noisy Student” with other TL models for two loss functions i.e., LR 5e-06 and LR 5e-03

Models [LR 5e-06]	Sensitivity	Specificity	Precision	F1-Score	Accuracy	MCC
AFD-Net-Noisy Student	1	0.99	0.97	0.89	[0.99, 0.98, 0.98, 0.97]	0.94
AFD-Net-ImageNet	0.99	0.99	0.96	0.88	[0.98, 0.97, 0.97, 0.96]	0.93
AFD-Net CutMixUp-ImageNet	0.98	0.98	0.96	0.87	[0.98, 0.96, 0.97, 0.96]	0.92
AFD-Net Dropout-ImageNet	0.99	0.99	0.96	0.88	[0.98, 0.98, 0.97, 0.97]	0.93
EfficientNet B3-ImageNet	0.98	0.98	0.95	0.85	[0.98, 0.96, 0.95, 0.95]	0.90
EfficientNet B4-ImageNet	0.98	0.98	0.96	0.86	[0.98, 0.96, 0.96, 0.97]	0.91
[InceptionV3: LR 5e-06]	0.99	0.99	0.95	0.87	[0.98, 0.97, 0.96, 0.96]	0.92
[ResNet50: LR 5e-06]	0.99	0.99	0.94	0.86	[0.98, 0.96, 0.96, 0.95]	0.91
[ResNet101: LR 5e-06]	0.98	0.99	0.95	0.86	[0.98, 0.96, 0.96, 0.96]	0.91
VGG16: LR 5e-06]	0.98	0.99	0.93	0.85	[0.97, 0.96, 0.96, 0.95]	0.90
Models [LR 5e-03]	Sensitivity	Specificity	Precision	F1-Score	Accuracy	MCC
AFD-Net-Noisy Student	0.99	0.99	0.96	0.88	[0.98, 0.97, 0.97, 0.96]	0.93
AFD-Net-ImageNet	0.98	0.98	0.95	0.87	[0.97, 0.96, 0.97, 0.95]	0.92
AFD-Net CutMixUp-ImageNet	0.98	0.97	0.94	0.86	[0.97, 0.96, 0.96, 0.95]	0.91
AFD-Net Dropout-ImageNet	0.99	0.99	0.96	0.88	[0.98, 0.98, 0.97, 0.97]	0.92
EfficientNet B3-ImageNet	0.98	0.98	0.95	0.85	[0.98, 0.96, 0.95, 0.96]	0.90
EfficientNet B4-ImageNet	0.98	0.98	0.96	0.86	[0.98, 0.95, 0.96, 0.96]	0.90
[InceptionV3: LR 5e-03]	0.99	0.98	0.94	0.85	[0.97, 0.96, 0.95, 0.96]	0.90
[ResNet50: LR 5e-03]	0.97	0.98	0.93	0.84	[0.97, 0.95, 0.95, 0.95]	0.88
[ResNet101: LR 5e-03]	0.98	0.98	0.94	0.85	[0.97, 0.96, 0.96, 0.96]	0.90
VGG16: LR 5e-03]	0.98	0.98	0.92	0.84	[0.97, 0.96, 0.95, 0.95]	0.89

6 Discussion and Conclusion

The world around us relies heavily on the agricultural sector to provide food. Early detection of plant diseases is critical to the industry. In this article, a AFD-Net model is proposed to identify leaf disease in apple trees. The proposed model is applied to two data sets: Plant Pathology 2020 and Plant Pathology 2021. The proposed AFD-Net model achieves 98.7% accuracy, which is higher than other TL models (B3, B4, Inception V3, VGG16, ResNet50, and 101). In addition, the performance of the proposed model outperforms the other models for both datasets. The above results show the efficiency of the proposed model in identifying leaf diseases on apple trees for major and minor classes, i.e., multiclassification.

In the future, a web-based application could be developed so that the farmer can use it directly on an image instead of a leaf. Accuracy could be improved by using computer vision based techniques such as segmentation and region-of-interest for detection and classification, which could lead to better predictions.

References

1. Ranjita Thapa, Kai Zhang, Noah Snavely, Serge Belongie, and Awais Khan. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Applications in Plant Sciences*, 8(9):e11390, 2020.
2. Mohit Agarwal, Rohit Kumar Kaliyar, Gaurav Singal, and Suneet Kr Gupta. Fcnn-lda: A faster convolution neural network model for leaf disease identification on apple’s leaf dataset. In *The International Conference on Information & Communication Technology and System (ICTS)*, pages 246–251, 2019.
3. Jayme Garcia Arnal Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46–53, 2018.
4. Alieu MOININA, Rachid LAHLALI, and Mohammed BOULIF. Important pests, diseases and weather conditions affecting apple production in morocco: Current state and perspectives. *Revue Marocaine des Sciences Agronomiques et Vétérinaires*, 7(1), 2019.
5. Yong Zhong and Ming Zhao. Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*, 168:105146, 2020.
6. Sammy V Militante, Bobby D Gerardo, and Nanette V Dionisio. Plant leaf detection and disease recognition using deep learning. In *IEEE Eurasia Conference on IOT, Communication and Engineering*, pages 579–582, 2019.
7. Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.
8. Alieu Moinina, Rachid Lahlali, Dustin MacLean, and Mohammed Boulif. Farmers’ knowledge, perception and practices in apple pest management and climate change in the fes-meknes region, morocco. *Horticulturae*, 4(4):42, 2018.
9. Anne-Katrin Mahlein. Plant disease detection by imaging sensors—parallels and specific demands for precision agriculture and plant phenotyping. *Plant Disease*, 100(2):241–251, 2016.
10. Jayme Garcia Arnal Barbedo. An automatic method to detect and measure leaf disease symptoms using digital image processing. *Plant Disease*, 98(12):1709–1716, 2014.

11. Pengbo Dai, Xiaofei Liang, Yajing Wang, Mark L Gleason, Rong Zhang, and Guangyu Sun. High humidity and age-dependent fruit susceptibility promote development of trichothecium black spot on apple. *Plant Disease*, 103(2):259–267, 2019.
12. Thomas Wöhner and Ofere Francis Emeriewen. Apple blotch disease (marssonina coronaria (ellis & davis) davis)—review and research prospects. *European journal of plant pathology*, 153(3):657–669, 2019.
13. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, 2016.
14. Sachin B Jadhav, Vishwanath R Udipi, and Sanjay B Patil. Identification of plant diseases using convolutional neural networks. *International Journal of Information Technology*, pages 1–10, 2020.
15. Jihen Amara, Bassem Bouaziz, and Alsayed Algergawy. A deep learning-based approach for banana leaf diseases classification. *Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband*, 2017.
16. Xiaofei Chao, Guoying Sun, Hongke Zhao, Min Li, and Dongjian He. Identification of apple tree leaf diseases based on deep learning models. *Symmetry*, 12(7):1065, 2020.
17. Hee-Jin Yu, Chang-Hwan Son, and Dong Hyuk Lee. Apple leaf disease identification through region-of-interest-aware deep convolutional neural network. *Journal of Imaging Science and Technology*, 64(2):20507–1, 2020.
18. Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *The International Conference on Natural Computation*, volume 4, pages 192–201, 2008.
19. Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *The IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.
20. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
21. Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.
22. Hamail Ayaz, Erick Rodríguez-Esparza, Muhammad Ahmad, Diego Oliva, Marco Pérez-Cisneros, and Ram Sarkar. Classification of apple disease based on non-linear deep features. *Applied Sciences*, 11(14):6422, 2021.
23. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
24. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.